BAB I

PENDAHULUAN

A. Latar belakang

Internet of Things (IoT) merupakan suatu konsep mengenai benda dalam kehidupan sehari-hari dengan sensor, *software*, atau teknologi lain di dalamnya, memungkinkannya terhubung dengan perangkat serta sistem lain melalui jaringan internet. Sederhananya, konsep ini bertujuan untuk menghubungkan objek fisik ke internet.

Banyak manfaat *Internet of Things* (IoT) yang bisa kita rasakan pada era *industry* 4.0 sekarang ini, misalnya dalam efisiensi waktu, meminimalisir *effort* kerja, meningkatkan proses *collecting* data, mengefektifkan analisis data, meningkatkan sistem keamanan dan manfaat-manfaat lainnya.

Pada *Internet of Things* (IoT) ada beberapa susunan unsur, seperti sensor, koneksi, pengolahan data serta *user interface*. Semua unsur yang menyusun ini saling terkait. Sensor mengumpulkan data dari lingkungan sekitar di mana benda ini berada, unsur koneksi mengirimkan data atau informasi dari sensor ke pengolahan data kemudian akan ditampilkan pada *user interface* yang berfungsi memberikan informasi kepada pengguna serta unsur *user interface* ini juga dapat berperan sebagai *controller*.

Implementasi IoT sudah banyak diterapkan di berbagai macam bidang, seperti bidang keamanan, bidang medis, bidang properti, bidang industri manufaktur dan lain-lain. Dulu pada awalnya, mesin dibuat hanya untuk membantu

manusia dan dioperasikan secara manual. Namun, seiring perkembangan teknologi yang begitu pesat, mesin bisa berjalan sendiri atau otomatis, tetapi dalam pemanfaatan mesin sebagai alat dalam sebuah sistem akan menemui kendala jika sudah menyangkut jarak dan waktu. Dengan jarak yang begitu jauh maka mesin tidak akan bisa berinteraksi dengan mesin lain, untuk mengatasi hal inilah implementasi IoT diterapkan pada sebuah mesin dengan pengenal IP *address* yang dapat menggunakan jaringan internet sebagai media komunikasi saling bertukar data.

Pemberian manfaat impelementasi IoT begitu besar. Karena hal tersebut penulis ingin mengangkat penelitian dalam implementasi IoT. Dalam penelitian nantinya, penulis menggunakan 4 (empat) sensor sebagai IoT perangkat modulnya. Masing-masing sensor akan mengirimkan data ke sistem *database* yang akan diteruskan ke situs web untuk dipantau secara *realtime* oleh pengguna di mana saja dan kapan saja selama terkoneksi dengan jaringan internet. Serta dalam penelitian ini, penulis nantinya akan menguji seberapa cepat kecepatan pengiriman data sensor ke *database* hingga ditampilkan pada situs web jika menggunakan 4 (empat) sensor atau multi sensor pada pengolahan datanya dalam pengimplementasian IoT.

Dengan demikian berdasarkan latar belakang masalah di atas, maka melalui tugas akhir ini penulis mengambil penelitian dengan judul "**Implementasi IoT sebagai Alat Pengolah Multi Sensor**". Diharapkan melalui penelitian ini dapat menjadi referensi pengimplementasian IoT dengan menggunakan multi sensor pada beberapa bidang implementasi IoT.

B. Rumusan masalah

Rumusan masalah pada penelitian ini adalah bagaimana mengimplementasikan IoT sebagai alat pengolah multi sensor?

C. Batasan masalah

Untuk menghindari permasalahan agar tidak meluas, maka penulis dalam hal ini membuat batasan permasalahan sebagai berikut:

1. Sistem implementasi IoT yang dibuat hanya memantau informasi yang ditampilkan pada situs web.
2. Sistem hanya menggunakan 4 (empat) sensor yakni sensor kekeruhan, sensor kadar pH, sensor warna dan sensor suhu.

D. Tujuan penelitian

1. Membuat sistem implementasi IoT sebagai alat pengolah multi sensor.
2. Menguji kecepatan pengiriman data sensor ke *database* hingga ditampilkan pada situs web.

E. Manfaat penelitian

1. Bagi penulis

   Dapat menambah wawasan tentang *Internet of Things* (IoT) dengan lebih mendalam termasuk pada implementasi multi sensor pada *Internet of Things* (IoT).

2. Bagi mahasiswa

   Dapat menjadi referensi pengimplementasian IoT multi sensor pada beberapa bidang implementasi IoT.

F. Sistematika penulisan

Sistematika penulisan dalam tugas akhir ini sesuai dengan acuan yang ada.

BAB I PENDAHULUAN

Berisi mengenai latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini berisi uraian-uraian tentang teori-teori yang menjadi landasan penelitian untuk pemecahan masalah.

BAB III METODE PENELITIAN

Membahas tentang waktu dan tempat penelitian, alat dan bahan yang digunakan dalam penelitian, serta langkah-langkah penelitian.

BAB IV HASIL PENELITIAN DAN PEMBAHASAN

Berisi mengenai hasil penelitian dan pembahasannya. Penyajian hasil penelitian memuat deskripsi sistematik tentang narasi. Dan tujuan pembahasan adalah menjawab pernyataan penelitian atau rumusan masalah, menafsirkan temuan-temuan, mengintegrasikan hasil dan temuan ilmu atau teori baru serta menjelaskan implikasi hasil penelitian.

BAB V PENUTUP

Pada bab penutup ini berisi berupa kesimpulan dan saran. Kesimpulan merangkum semua hasil penelitian yang telah diuraikan. Dan saran berdasarkan hasil penelitian dan pengalaman serta pertimbangan peneliti yang ditujukan kepada para peneliti lain yang akan melanjutkan atau mengembangkan penelitian lebih lanjut.

BAB II

TINJAUAN PUSTAKA

A. Kajian teori

1. *Internet of Things* (IoT)



Gambar 2.1 Ilustrasi *Internet of Things*
Sumber: (https://etn.co.id)

Internet of Things (IoT) tengah menjadi topik pembicaraan yang semakin hangat di era revolusi industri 4.0 ini lantaran konsepnya yang tidak hanya memiliki potensi untuk mempengaruhi *lifestyle* kita tetapi juga bagaimana semua bekerja.

Internet of Things (IoT) adalah konsep komputasi tentang objek sehari-hari yang terhubung ke internet dan mampu mengidentifikasi diri ke perangkat lain.

Menurut metode identifikasi RFID (*Radio Frequency Identification*), istilah IoT tergolong dalam metode komunikasi, meskipun IoT juga dapat

mencakup teknologi sensor lainnya, teknologi nirkabel atau kode QR (*Quick Response*).

Internet of Things (IoT) sebenarnya adalah konsep yang cukup sederhana, yang artinya menghubungkan semua objek fisik di kehidupan sehari-hari ke internet.

Dalam *Internet of Things* (IoT), ada beberapa unsur yang menyusunnya, yaitu:

a. Sensor

Unsur pertama adalah sensor. Hal ini bertugas untuk mengumpulkan data dari lingkungan sekitar di mana benda tersebut berada. Sensor tersebut bisa mengumpulkan data berupa suhu, cahaya, suara, waktu, dan lain-lain.

b. Koneksi

Unsur ini bertugas untuk mengirimkan data atau informasi dari sensor ke pengolahan data. Pada konsep ini, jaringan internet berperan besar dalam hal koneksi. Selain itu, konsep ini juga bisa memanfaatkan koneksi dari *wifi* atau *bluetooth*.

c. Pengolahan data

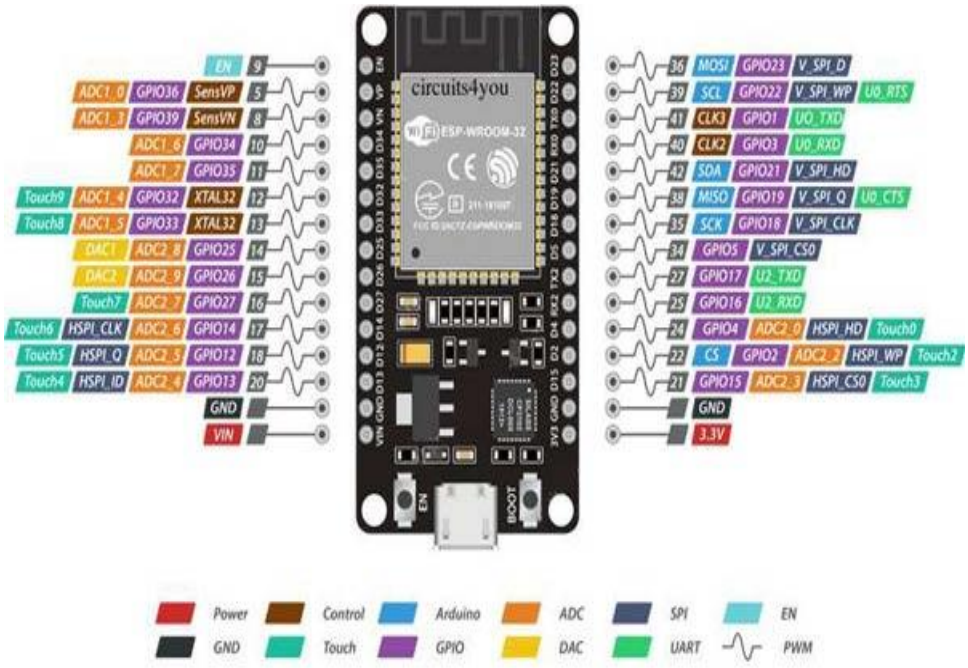
Unsur inilah yang akan menilai suatu data. Bagaimana suatu perangkat akan beroperasi, mengacu pada hal ini. Misalnya pada lampu pintar. Pengolahan data akan menilai kapasitas cahaya. Jika cahaya sudah redup, lampu akan menyala.

d. *User interface*

Unsur ini dapat berperan sebagai *controller*. Selain itu, perangkat *user interface* juga berfungsi untuk memberikan informasi *user*. Perangkat *user interface* bisa terpasang jadi satu dengan barang, bisa juga terpisah. Perangkat *user interface* juga bisa menggunakan *smartphone* dengan perantara suatu aplikasi.

2. NodeMCU Esp32

ESP32 adalah mikrokontroler yang dikenalkan oleh *Espressif System* merupakan penerus dari mikrokontroler ESP8266. Pada mikrokontroler ini sudah tersedia modul *wifi* dalam chip sehingga sangat mendukung untuk membuat sistem aplikasi *Internet of Things* (IoT). Perlu diketahui bahwa ESP32 ini memiliki 2 (dua) tegangan operasi yakni 5V dan 3.3V. Dengan pin keluaran terdiri dari 18 ADC (Analog Digital *Converter*, berfungsi untuk merubah sinyal analog ke digital), 2 DAC (Digital Analog *Converter*, kebalikan dari ADC), 16 PWM (*Pulse Width Modulation*), 10 sensor sentuh, 2 (dua) jalur antarmuka UART, dan pin antarmuka I2C, I2S, dan SPI.



Gambar 2.2 NodeMCU Esp32
Sumber: (lms.onnocenter.or.id)

Spesifikasi nodeMCU Esp 32:

a. Prosesor: *Tensilica Xtensa* 32-bit LX6 *microprocessor*. Frekuensi untuk *clock* maksimun 240 MHz.

b. Konektivitas nirkabel: *Wifi* 802.11 b/g/n/e/i (802.11n @2.4 GHz hingga 150 Mbit/s). *Bluetooth* v4.2 BR/EDR dan *Bluetooth Low Energy* (BLE).

c. Memori *internal*: ROM 448 KB, SRAM 520 KB, RTC fast SRAM 8 KB, RTC slow SRAM 8 KB, eFuse 1 Kbit.

d. *Flash* tertanam: 0 MB (ESP32-D0WDQ6, ESP32-D0WD, dan chip ESP32-S0WD), 2 MB (chip ESP32-D2WD), 4 MB (modul ESP32-PICO-D4 SiP).

e. *Flash eksternal* & SRAM: Memori *flash eksternal* sebesar 16 MB yang dipetakan ke dalam ruang kode pada CPU, Esp32 juga telah mendukung akses data 8-bit, 16-bit dan 32-bit.

f. Memori *flash* atau SRAM *eksternal*: Sebesar 8 MB yang dipetakan ke dalam ruang data CPU, Esp32 telah mendukung akses data 8-bit, 16-bit dan 32-bit. Pembacaan data didukung pada memori flash dan SRAM. Sedangkan penulisan data didukung pada memori SRAM.

g. I/O: Jumlah pin : 30 meliputi pin tegangan dan GPIO, 15 pin ADC (Analog *to* Digital *Converter*), 3 UART Interface, 3 SPI Interface, 2 I2C Interface, 16 pin PWM (*Pulse Width Modulation*), 2 pin DAC (Digital *to* Analog *Converter*).

h. *Security*: IEEE 802.11 standard *security features all supported*, *including* WFA, WPA/WPA2 dan WAP. 1024-bit OTP, up to 768-bit *for customers*. *Cryptographic hardware acceleration* AES, SHA-2, RSA, *elliptic curve cryptography* (ECC), *random number generator* (RNG).

3. Kekeruhan

Kekeruhan menggambarkan sifat optik air yang ditentukan berdasarkan banyaknya cahaya yang diserap dan dipancarkan oleh bahan-bahan yang terdapat di dalam air. Kekeruhan disebabkan adanya bahan organik dan anorganik yang tersuspensi dan terlarut (misalnya lumpur dan pasir halus), maupun bahan organik dan anorganik yang berupa plankton dan mikroorganisme lain (APHA, 1976; Davis dan Cornwell, 1991 dalam Effendi 2003). Zat anorganik yang menyebabkan kekeruhan dapat berasal dari

pelapukan batuan dan logam, sedangkan zat organik berasal dari lapukan hewan dan tumbuhan. Bakteri dapat dikategorikan sebagai materi organik tersuspensi yang menambah kekeruhan air.

Padatan tersuspensi berkolerasi positif dengan kekeruhan. Semakin tinggi nilai padatan tersuspensi, semakin tinggi nilai kekeruhan. Akan tetapi, tingginya padatan terlarut tidak selalu diikuti dengan tingginya kekeruhan. Tingginya nilai kekeruhan dapat mempersulit usaha penyaringan dan mengurangi efektivitas desinfeksi pada proses penjernihan air.

Secara optis, kekeruhan merupakan suatu kondisi yang mengakibatkan cahaya dalam air didispersikan atau diserap dalam suatu contoh air.

4. *Turbidity* sensor DFRobot

Turbidity sensor merupakan sensor yang berfungsi mengukur kualitas air dengan mendeteksi tingkat kekeruhannya. Sensor ini mendeteksi partikel tersuspensi dalam air dengan cara mengukur transmitansi dan hamburan cahaya yang berbanding lurus dengan kadar Total *Suspended Solids* (TTS). Semakin tinggi kadar TTS, maka semakin tinggi pula tingkat kekeruhan air tersebut. Dalam kekeruhan air, satuan ukuran yang dipakai adalah NTU atau *Nephelometric Turbidity Unit*.

Sensor ini mendukung dua mode *output*, digital dan analog sehingga dapat dengan mudah diakses melalui arduino atau mikrokontroller lainnya. sensor ini dapat diaplikasikan untuk mengukur tingkat kekeruhan air pada sungai, danau, laboratorium, limbah cair, dsb.



Gambar 2.3 *Turbidity* Sensor DFRobot
Sumber: (https://digiwarestore.com)

Spesifikasi *turbidity* sensor DFRobot:

a. Tegangan Operasi: 5V DC.
b. Operasi aliran: 40mA (*max*).
c. Waktu Respon : <500ms.
d. Resistansi Isolasi: 100M (Min).
e. Metode Keluaran:
    1) *Output* analog: 0-4.5V.
    2) *Output* Digital: Sinyal level Tinggi/Rendah (Anda dapat menyesuaikan nilai ambang dengan menyesuaikan potensiometer).
f. Suhu Operasi: 5 ~ 90.
g. Suhu Penyimpanan: -10 ~ 90.
h. Berat: 30g.
i. Dimensi Adaptor: 38mm x 28mm x 10mm/1.5 inci x 1.1 inci x 0.4 inci.

5. Kadar pH

PH air adalah sebuah satuan ukur yang mengukur kadar keasaman atau kebasaan air. Rentang ukur pH air dimulai dari 0 hingga 14. PH air dengan 0 sampai 6 memiliki kadar keasaman, sedangkan pH air dari 8 sampai 14

memiliki kadar kebasaan. PH air 7 merupakan kadar air netral dan sering digunakan untuk minum.

Arti pH yang sebenarnya adalah ukuran jumlah relatif ion hidrogen dan hidroksil yang bebas. Karena kandungan pH dapat dipengaruhi oleh bahan kimia, maka pH merupakan indikator penting dalam air yang berubah secara kimia. Setiap angka mewakili perubahan 10 kali lipat dalam keasaman/kebasaan. Air dengan pH lima adalah sepuluh kali liipat lebih asam dari air yang memiliki pH enam.

Dalam kehidupan sehari-hari, kandungan pH terhadap air sangat besar. Air yang memiliki pH air terlalu rendah akan terasa pahit / asam, sedangkan jika air memiliki pH terlalu tinggi maka air berasa tidak enak (kental/licin). Selain itu, salah satu fungsi air bagi manusia adalah mengeluarkan racun dari dalam tubuh, air dengan kandungan pH netral dapat membantu produksi darah netral sehingga mudah mengeluarkan racun dari tubuh. Ahli kesehatan merekomendasikan untuk mengkonsumsi pH air yang berada di antara 6,5 – 8,5.

6. Sensor pH 4502C

Merupakan kit sensor pengukur tingkat keasaman pada cairan (pH) yang menggunakan elektroda berstandar industri sebagai komponen penginderaan utamanya. Elektroda pada sensor ini terbuat dari membran kaca sensitif dengan impendasi kecil sehingga dapat menghasilkan pengukuran dengan respon yang cepat serta kestabilan terhadap suhu yang tinggi. Hasil pembacaan sensor dapat langsung diakses oleh mikrokontroler melalui

antarmuka pH 2.0 yang terdapat pada sensor. Sensor ini sangat ideal untuk aplikasi pengukuran pH cairan dalam jangka panjang.



Gambar 2.4 Sensor pH 4502C
Sumber: (https://store.roboticsbd.com)

Spesifikasi sensor pH 4502C:

a. Tegangan pemanas: 5±0.2V dan 3.3V(AC - DC).
b. *Working current*: 5-10mA.
c. Rentang konsentrasi deteksi: PH 0-14.
d. Rentang deteksi suhu: 0-80 °C.
e. Waktu respons: 5 detik.
f. Waktu stabilitas: 120 detik.
g. Konsumsi daya: 0.5W.
h. Suhu kerja: -10 ~ 50 celcius (suhu nominal 20 celcius).
i. Kelembaban kerja: 95% RH (kelembaban nominal 65% RH).
j. Kehidupan pelayanan: 3 tahun.
k. Ukuran: 42mm x 32mm x 20mm.
l. Berat: 25g.

m. *Output*: *output* sinyal tegangan analog.

7. Warna air

Air minum sebaiknya tidak berwarna untuk alasan estetika dan untuk mencegah keracunan dari berbagai zat kimia maupun mikroorganisme yang berwarna. Warna dapat menghambat penetrasi cahaya ke dalam air. Warna pada air disebabkan oleh adanya partikel hasil pembusukan bahan organik, ion-ion metal.

Alam (besi dan mangan), plankton, humus, buangan industri, dan tanaman air. Adanya oksida besi menyebabkan air berwarna kemerahan, sedangkan oksida mangan menyebabkan air berwarna kecoklatan atau kehitaman. Kadar besi sebanyak 0,3 mg/l dan kadar mangan sebanyak 0,05 mg/l sudah cukup dapat menimbulkan warna pada perairan (*peavy et al.*, 1985 dalam Effendi, 2003). Kalsium karbonat yang berasal dari daerah berkapur menimbulkan warna kehijauan pada perairan. Bahan-bahan organik, misalnya tanin, lignin, dan asam humus yang berasal dari dekomposisi tumbuhan yang telah mati menimbulkan warna kecoklatan.

8. Sensor warna TCS230

Sensor warna TCS230 adalah sensor warna yang sering digunakan pada aplikasi mikrokontroler untuk pendeteksian suatu objek benda atau warna dari objek yang dipantau. Sensor warna TCS230 juga dapat digunakan sebagai sensor gerak, di mana sensor mendeteksi gerakan suatu objek berdasarkan perubahan warna yang diterima oleh sensor. Pada dasarnya sensor warna TCS230 adalah rangkaian *photodioda* yang disusun secara *matrik array* 8x8

dengan 16 buah konfigurasi *photodioda* yang berfungsi sebagai filter warna merah, 16 *photodioda* sebagai filter warna biru dan 16 *photodioda* lagi tanpa filter warna. Sensor warna TCS230 merupakan sensor yang dikemas dalam chip DIP 8 pin dengan bagian muka transparan sebagai tempat menerima intensitas cahaya yang berwarna.



Gambar 2.5 Sensor warna TCS230
Sumber: (https://beetrona.com)

Spesifikasi sensor warna TCS230:

a. Tegangan *input* 3V ~ 5V.
b. 16 *photodiode* untuk memfilter warna merah.
c. 16 *photodiode* untuk memfilter warna hijau.
d. 16 *photodiode* untuk memfilter warna biru.
e. 16 *photodioda* tanpa filter.
f. Pin S0 - S1: Pin untuk menyeleksi *input* frekuensi *output.*
g. Pin S2 – S3: Sensor masukan *photodioda*.
h. Pin OUT: keluaran frekuensi.
i. Pin OE: aktifkan pin *output* (aktif *low*).
j. Ukuran: 31x23x19mm.

k. Berat: 4g.

9. Suhu air

Suhu air sebaiknya sejuk atau tidak panas, agar tidak terjadi pelarutan zat kimia pada saluran/pipa yang dapat membahayakan kesehatan, menghambat reaksi-reaksi biokimia di dalam saluran/pipa, mikroorganisme patogen tidak mudah berkembang biak, dan bila diminum dapat menghilangkan dahaga.

Suhu suatu badan air dipengaruhi oleh musim, lintang (*latitude*), ketinggian dari permukaan laut (*altitude*), waktu, sirkulasi udara, penutupan awan, aliran, serta kedalaman. Perubahan suhu mempengaruhi proses fisika, kimia, dan biologi badan air. Suhu berperan dalam mengendalikan kondisi ekosistem perairan.

Peningkatan suhu mengakibatkan peningkatan viskositas, reaksi kimia, evaporasi, *volatilisasi*, serta menyebabkan penurunan kelarutan gas dalam air (gas O2, CO2, N2, CH4, dan sebagainya) (Haslam, 1995 dalam Effendi, 2003). Peningkatan suhu juga menyebabkan terjadinya peningkatan dekomposisi bahan organik oleh mikroba. Kisaran suhu optimal bagi pertumbuhan fitoplankton di perairan adalah 20 ^{o}C – 30 ^{o}C.

10. Sensor suhu DS18B20

Sensor suhu DS18B20 adalah sensor suhu yang menggunakan *interface one wire*, sehingga hanya menggunakan kabel yang sedikit dalam instalasinya. Uniknya sensor ini bisa dijadikan paralel untuk satu *input*. Artinya kita bisa menggunakan sensor DS18B20 lebih dari satu namun keluarannya hanya

dihubungkan ke satu pin arduino. Alasan ini membuat banyak sensor digunakan, apalagi sensor ini memiliki tipe *waterproof*, sehingga sensor ini bisa digunakan sebagai alat ukur air maupun kontrol pemanas air.



Gambar 2.6 Sensor suhu DS18B20
Sumber: (https://lh3.googleusercontent.com)

Spesifikasi sensor suhu DS18B20:

a. Berkomunikasi menggunakan metode *One Wire.*
b. Tegangan operasi: 3V hingga 5V.
c. Rentang Suhu: -55 ° C hingga + 125 ° C.
d. Akurasi: ± 0,5 ° C.
e. Resolusi *Output*: 9-bit hingga 12-bit (dapat diprogram).
f. Alamat 64-bit yang unik memungkinkan *multiplexing*.
g. Waktu konversi: 750ms pada 12-bit.
h. Opsi alarm yang dapat diprogram.
i. Tersedia sebagai To-92, SOP dan bahkan sebagai sensor tahan air.

11. HTML (*Hypertext Markup Language*)

Hypertext Markup Language atau biasa disebut HTML adalah bahasa standar pemrograman untuk membuat suatu situs web yang bisa diakses dengan internet. Dengan kata lain halaman situs web yang kita lihat dan kita baca disusun dengan menggunakan bahasa ini dan kemudian diterjemahkan oleh komputer agar dapat dipahami oleh penggunanya. HTML merupakan standar pembuatan situs web secara luas agar laman situs web dapat ditampilkan pada layar komputer.

HTML disusun dengan kode dan simbol tertentu yang dimasukkan ke dalam sebuah file atau dokumen. Jadi setiap Anda membuka situs web apa pun dengan menggunakan *browser* maka situs web tersebut dibuat dengan menggunakan HTML.

Dari singkatan *hypertext markup language* itu sendiri kita bisa mengetahui makna dari HTML. *Hypertext* adalah suatu metode yang digunakan untuk berpindah laman situs web ke laman yang lain dengan mengklik suatu tulisan atau simbol pada laman situs web.

Istilah *markup* pada HTML sendiri diartikan sebagai suatu hal yang dilakukan *tag* HTML terhadap teks yang berada di dalamnya, misal jika mengetik suatu teks dengan tanda *tag* <b> maka teks tersebut akan muncul sebagai teks dengan huruf tebal atau *bold* pada suatu laman situs web.

Simbol atau *tag* HTML tersebut ditulis pada laman HTML yang sudah disediakan pada *dashboard* situs web pada umumnya. Sementara *language* dalam HTML adalah bahasa pemrograman atau *script* yang disusun dari *tag*-

tag tertentu yang nantinya akan diterjemahkan ke dalam teks atau visual yang dapat dilihat pada situs web.

12. *JavaScript*

JavaScript adalah bahasa pemrograman komputer yang dinamis. Pada umumnya *JavaScript* digunakan pada web *browser* untuk menciptakan halaman web yang menarik, interaktif serta menerapkan berbagai fungsi pada halaman web.

Kegunaan utama *JavaScript* adalah untuk menuliskan fungsi yang disisipkan ke dalam HTML, baik disisipkan secara langsung maupun diletakan ke dalam *file* teks dan di *link* dari dokumen HTML. Secara fungsional, *JavaScript* dipakai untuk menyediakan akses *script* pada objek yang dibenamkan (*embedded*).

Contoh penggunaan *JavaScript*, yaitu merubah *image* kursor ketika melewati objek tertentu, fungsi validasi pada *form* sebelum data dikirimkan ke *server* dan sebagainya.

13. PHP (*Hypertext Preprocessor*)

PHP adalah bahasa pemrograman *script server-side* yang didesain untuk pengembangan web. Selain itu, PHP juga bisa digunakan sebagai bahasa pemrograman umum. PHP dikembangkan pada tahun 1995 oleh Rasmus Lerdorf, dan sekarang dikelola oleh The PHP Group.

PHP disebut bahasa pemrograman *server side* karena PHP diproses pada komputer *server*. Hal ini berbeda dibandingkan dengan bahasa

pemrograman *client-side* seperti *JavaScript* yang diproses pada web *browser* (*client*).

Pada awalnya PHP merupakan singkatan dari *Personal Home Page.* Sesuai dengan namanya, PHP digunakan untuk membuat situs web pribadi. Dalam beberapa tahun perkembangannya, PHP menjelma menjadi bahasa pemrograman web yang *powerfull* dan tidak hanya digunakan untuk membuat halaman web sederhana, tetapi juga situs web populer yang digunakan oleh jutaan orang seperti *wikipedia*, *wordpress*, *joomla*, dan lain sebagainya.

Saat ini PHP adalah singkatan dari PHP: *Hypertext Preprocessor*, sebuah kepanjangan rekursif, yakni permainan kata di mana kepanjangannya terdiri dari singkatan itu sendiri: PHP**:** *Hypertext Preprocessor*.

PHP dapat digunakan dengan gratis dan bersifat *open source*. PHP dirilis dalam lisensi PHP *license*, sedikit berbeda dengan lisensi GNU *General Public License* (GPL) yang biasa digunakan untuk proyek *open source*.

Untuk membuat halaman web, sebenarnya PHP bukanlah bahasa pemrograman yang wajib digunakan. Kita bisa saja membuat situs web hanya menggunakan HTML saja. Web yang dihasilkan dengan HTML (dan CSS) ini dikenal dengan website statik, dimana konten dan halaman web bersifat tetap.

Sebagai perbandingan, situs web dinamis yang bisa dibuat menggunakan PHP adalah situs web yang bisa menyesuaikan tampilan konten tergantung situasi. Situs web dinamis juga bisa menyimpan data ke dalam *database*, membuat halaman yang berubah-ubah sesuai *input* dari *user*, memproses *form*, dan lain-lain.

Untuk pembuatan web, kode PHP biasanya disisipkan ke dalam dokumen HTML. Karena fitur inilah PHP disebut juga sebagai *Scripting Language* atau bahasa pemrograman *script*.

14. *FireBase realtime database*

FireBase adalah suatu layanan Google yang digunakan untuk mempermudah para pengembang aplikasi dalam mengembangkan aplikasinya. Dengan adanya *firebase*, pengembang aplikasi bisa fokus mengembangkan aplikasi tanpa harus memberikan usaha yang besar. Dua fitur yang menarik dari *firebase* yaitu *Firebase Remote Config* dan *Firebase RealTime Database*.

Realtime Database adalah *database* NoSQL, sehingga memiliki pengoptimalan dan fungsionalitas yang berbeda dengan *database* terkait. API *Realtime Database* dirancang agar hanya mengizinkan operasi yang dapat dijalankan dengan cepat. Hal ini memungkinkan untuk membangun pengalaman *realtime* yang luar biasa dan dapat melayani jutaan pengguna tanpa mengorbankan kemampuan *respons*. Oleh karena itu, perlu dipikirkan bagaimana pengguna mengakses data, kemudian buat struktur data sesuai dengan kebutuhan tersebut. Berikut ini bagaimana proses penginstalan dan persiapan *firebase* ke aplikasi android:

a. Menghubungkan aplikasi ke *firebase*

Tambahkan *firebase* ke *project* android yang mana dalam *file build.gradle level project*, pastikan untuk memasukkan *repositori Maven* Google di bagian *buildscript* dan *allprojects*.

b. Menambahkan *Realtime Database* ke aplikasi

Menambahkan dependensi untuk *Realtime Database* ke *file build.gradle* level aplikasi dengan *script* aplikasi:

implementation 'com.google.firebase:firebase-database:19.1.0'

c. Mengonfigurasi aturan *Realtime Database*

Realtime Database menyediakan bahasa aturan deklaratif sehingga *programmer* dapat menentukan cara data disusun, diindeks, serta kapan data tersebut dapat dibaca dan ditulisi. Secara default, akses baca dan tulis ke *database* dibatasi, sehingga hanya pengguna yang terautentikasi yang dapat membaca atau menulis data. Untuk memulai tanpa menyiapkan autentikasi, serta dapat mengonfigurasi aturan agar dapat diakses oleh publik. Tindakan ini membuat *database* terbuka bagi siapa saja, termasuk orang yang tidak menggunakan aplikasi. Oleh karena itu, pastikan kembali untuk membatasi *database* setelah menyiapkan autentikasi.

d. Menulis ke *database*

Ambil *instance database* menggunakan *getInstance()* dan rujukan lokasi tempat dimana ingin menulis.

```
// Write a message to the database
FirebaseDatabase database = FirebaseDatabase.getInstance();
DatabaseReference myRef = database.getReference("message");

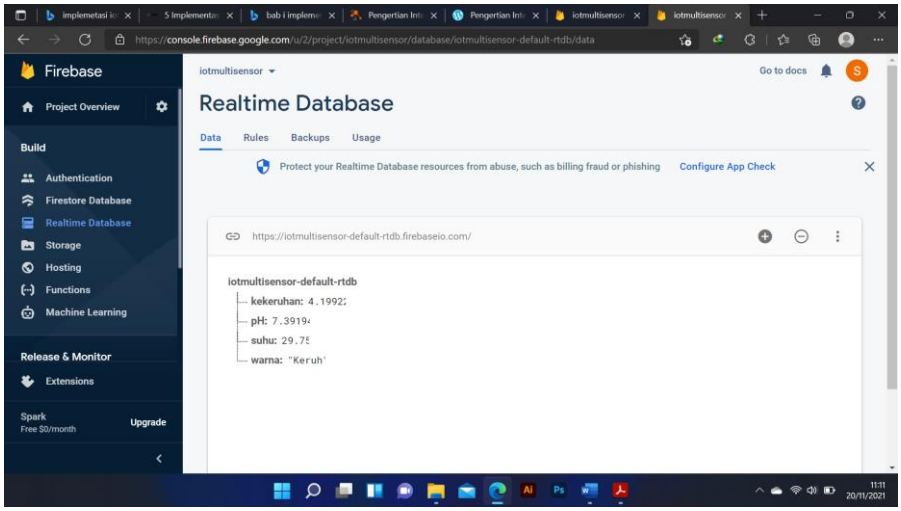

myRef.setValue("Hello, World!");
```

Sehingga bisa menyimpan berbagai jenis data ke *database* dengan cara ini, termasuk objek java. Ketika menyimpan sebuah objek, respon dari setiap *getter* akan disimpan sebagai turunan dari lokasi ini.

e. Membaca dari *database*

Agar data aplikasi di*update* secara *real-time*, maka perlu untuk menambahkan *ValueEventListener* ke referensi yang baru saja dibuat. Metode *onDataChange*() di *class* ini dipicu satu kali saat pemroses ditambahkan dan dipicu lagi setiap kali terjadi perubahan data, termasuk turunannya.

```
// Read from the database
myRef.addValueEventListener(newValueEventListener(){
    @Override
    publicvoid onDataChange(DataSnapshot dataSnapshot){
        // This method is called once with the initial value and again
        // whenever data at this location is updated.
        String value = dataSnapshot.getValue(String.class);
        Log.d(TAG,"Value is: "+ value);
    }
```



Gambar 2.7 Tampilan *firebase realtime database*

15. Bahasa pemprograman C

Bahasa Pemrograman C adalah sebuah bahasa pemrograman komputer yang bisa digunakan untuk membuat berbagai aplikasi (*general-purpose programming language*), mulai dari sistem operasi, seperti (Windows atau Linux) antivirus, *software* pengolah gambar (*image processing*), hingga *compiler* untuk bahasa pemrograman, dimana C banyak digunakan untuk membuat bahasa pemrograman lain yang salah satunya adalah PHP.

Meskipun termasuk *general-purpose programming language*, yakni bahasa pemrograman yang bisa membuat berbagai aplikasi, bahasa pemrograman C paling cocok merancang aplikasi yang berhubungan langsung dengan sistem operasi dan *hardware*. Ini tidak terlepas dari tujuan awal bahasa C dikembangkan.

Bahasa pemrograman C dibuat pertama kali oleh Dennis M. Ritchie pada tahun 1972. Saat itu Ritchie bekerja di Bell Labs, sebuah pusat penelitian yang berlokasi di Murray Hill, New Jersey, Amerika Serikat.

Ritchie membuat bahasa pemrograman C untuk mengembangkan sistem operasi UNIX. Sebelumnya, sistem operasi UNIX dibuat menggunakan bahasa *assembly* (*assembly language*). Akan tetapi bahasa *assembly* sendiri sangat rumit dan susah untuk dikembangkan.

Dengan tujuan mengganti bahasa *assembly*, peneliti di Bell Labs membuat bahasa pemrograman B. Namun bahasa pemrograman B juga memiliki beberapa kekurangan, yang akhirnya dilengkapi oleh bahasa pemrograman C.

Dengan bahasa C inilah sistem operasi UNIX ditulis ulang. Pada gilirannya, UNIX menjadi dasar dari banyak sistem operasi modern saat ini, termasuk Linux, Mac OS (iOS), hingga sistem operasi Android.

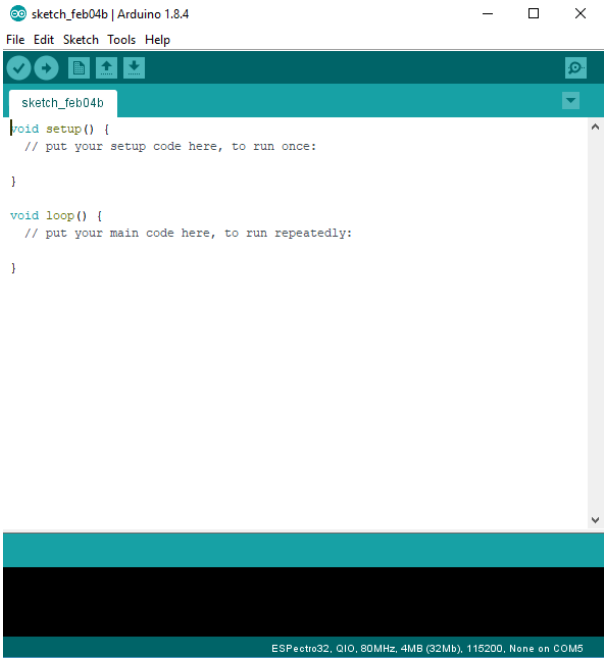
16. Arduino IDE

IDE merupakan singkatan dari *Integrated Development Enviroenment*. Secara harfiah dapat diartikan sebagai lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Arduino IDE adalah sebuah *software* untuk memprogram arduino. Pada *software* inilah arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang dibenamkan melalui *syntax* pemrograman. Arduino menggunakan bahasa pemrograman C yang dimodifikasi.

Bahasa pemrograman arduino sudah dirubah untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Didalam arduino sendiri sudah terdapat IC mikrokontroler yang sudah ditanami program yang

bernama *bootloader*. Funsi dari *bootloader* tersebut adalah untuk menjadi penengah antara *compiler* arduino dan *mikrokontroler*.

Arduino IDE dibuat dari bahasa pemrograman java yang dilengkapi dengan *library* C/C++ (*wiring*), yang membuat operasi *input*/*output* lebih mudah.



Gambar 2.8 Tampilan arduino IDE
Sumber: (https://miro.medium.com)

17. UML (*Unified Modelling Language*)

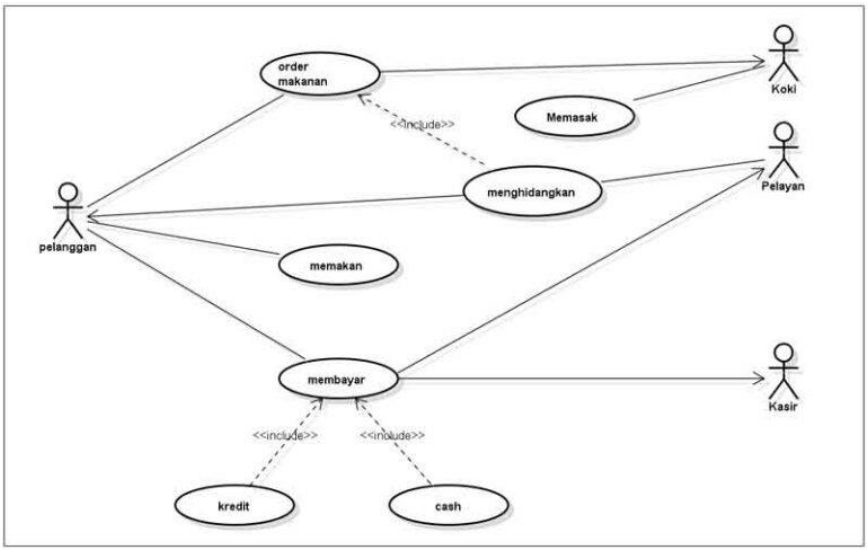
UML atau singkatan dari *Unified Modelling Language* merupakan suatu metode permodelan secara visual untuk sarana perancangan sistem berorientasi objek, atau dari definisi UML yaitu sebagai suatu bahasa yang sudah menjadi standar pada visualisasi, perancangan dan juga pendokumentasian sistem *software*.

UML digunakan untuk memodelkan suatu sistem (bukan hanya perangkat lunak) yang menggunakan konsep berorientasi objek. Dan juga untuk menciptakan suatu bahasa pemodelan yang dapat digunakan baik oleh manusia maupun mesin.

Adapun jenis-jenis diagram UML sebagai berikut:

a. *Use case* diagram

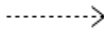
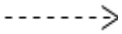
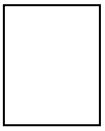
Adalah suatu urutan interaksi yang saling berkaitan antar sistem dan aktor. *Use case* dijalankan melalui melalui cara menggambarkan tipe interaksi antara user suatu program (sistem) dengan sistemnya sendiri.



Gambar 2.9 *Use case*
Sumber: (http://www.pengertianku.net)

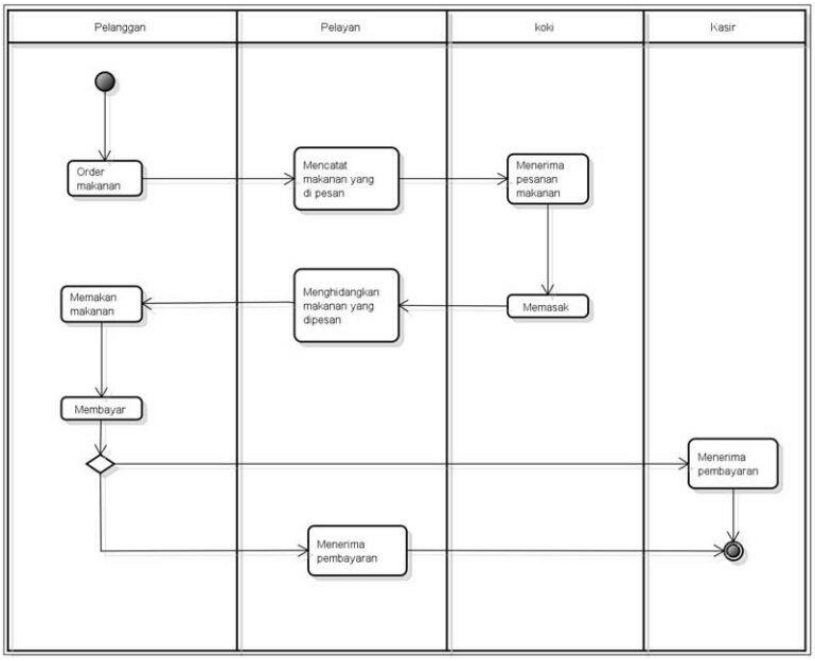
Tabel 2.1 Simbol *use case* diagram

No.	Gambar	Nama	Keterangan
1.		*Actor*	Spesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.

No.	Gambar	Nama	Keterangan
2.		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung pada elemen yang tidak mandiri.
3.		*Generalization*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dengan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
4.		*Include*	Menspesifikasikan bahwa *use case* sumber secara *eksplisit*.
5.		*Extend*	Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
6.		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7.		*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8.		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9.		*Collaboration*	Interaksi aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen.
10.		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

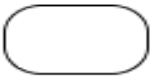
b. *Activity* diagram

Activity diagram atau diagram aktivitas yaitu salah satu jenis diagram pada UML yang dapat memodelkan proses-proses apa saja yang terjadi pada sistem.



Gambar 2.10 *Activity* diagram
Sumber: (http://pengertianku.net)

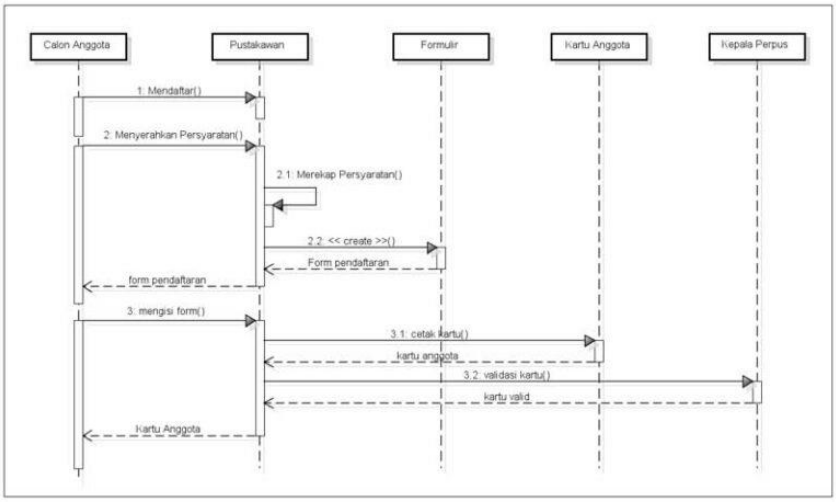
Tabel 2.2 Simbol *activity* diagram

No.	Gambar	Nama	Keterangan
1.		*Activity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2.		*Action*	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3.		*Initial Node*	Bagaimana objek dibentuk atau diawali.

No.	Gambar	Nama	Keterangan
4.		*Actifity Final Node*	Bagaimana objek dibentuk dan dihancurkan
5.		*Fork Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

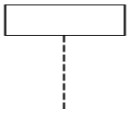
c. *Sequence* diagram

Sequence diagram yaitu yang menjelaskan interaksi objek yang berdasarkan urutan waktu, *sequence* diagram juga dapat menggambarkan urutan atau tahapan yang harus dilakukan untuk dapat menghasilkan sesuatu seperti pada *use case* diagram.



Gambar 2.11 *Sequence* diagram
Sumber: (http://pengertianku.net)

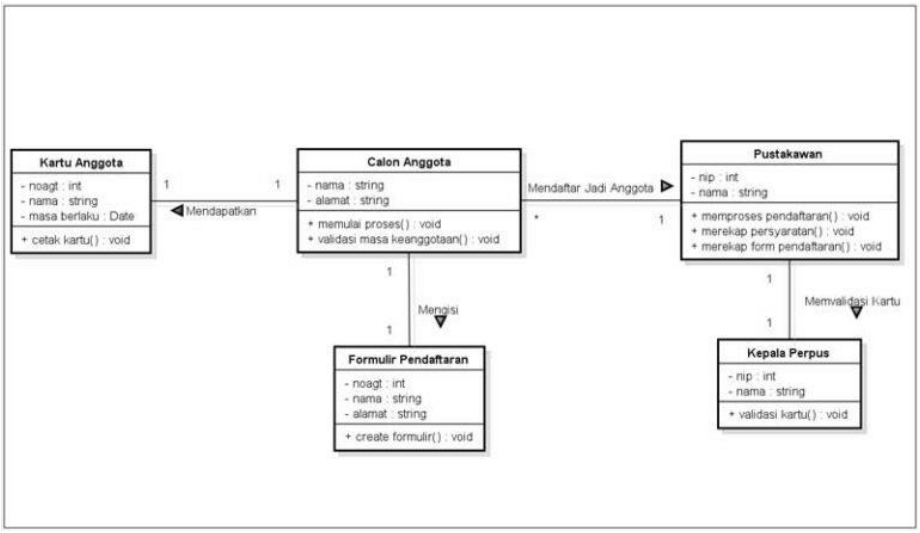
Tabel 2.3 Simbol *sequence* diagram

No.	Gambar	Nama	Keterangan
1.		*LifeLine*	Objek *entity*, antarmuka yang saling berinteraksi.

No.	Gambar	Nama	Keterangan
2.		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3.		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

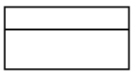
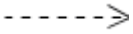
d. *Class* diagram

Class diagram yang digunakan untuk menampilkan kelas-kelas maupun paket-paket yang ada pada suatu sistem yang nantinya akan digunakan. Jadi, diagram ini dapat memberikan sebuah gambaran mengenai sistem maupun relasi-relasi yang terdapat pada sistem tersebut.



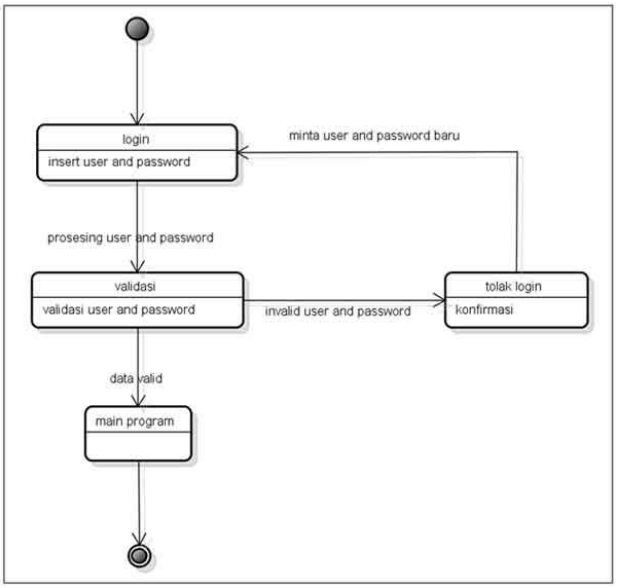
Gambar 2.12 *Class* diagram
Sumber: (http://pengertianku.net)

Tabel 2.4 Simbol *class* diagram

No.	Gambar	Nama	Keterangan
1.		*Generalization*	Hubungan di mana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
2.		*Nary Association*	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3.		*Class*	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4.		*Collaboration*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
5.		*Realization*	Operasi yang benar-benar dilakukan oleh suatu objek.
6.		*Dependency*	Hubungan di mana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
7.		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.

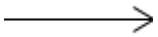
e. *Statemachine* diagram

Statemachine diagram yaitu salah satu jenis diagram pada UML yang menggambarkan transisi maupun perubahan keadaan suatu objek pada sistem.



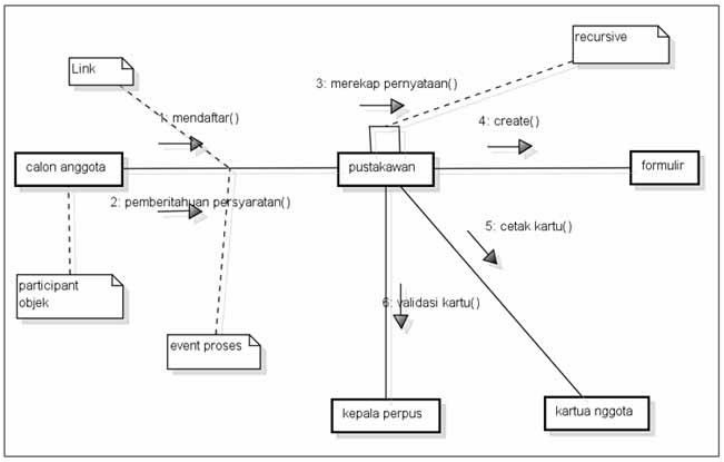
Gambar 2.13 *Statemachine* diagram
Sumber: (http://www.pengertianku.net)

Tabel 2.5 Simbol *statemachine* diagram

No.	Gambar	Nama	Keterangan
1.		*State*	Nilai atribut dan nilai *link* pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2.		*Initial Pseudo State*	Bagaimana objek dibentuk atau diawali.
3.		*Final State*	Bagaimana objek dibentuk dan dihancurkan.
4.		*Transition*	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya.
5.		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6.		*Node*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

f. *Communication* diagram

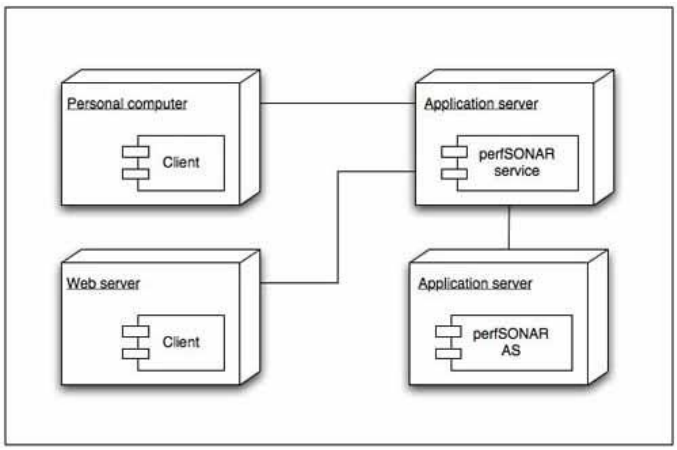
Salah satu jenis diagram pada UML yang dapat menggambarkan tahapan terjadinya suatu aktifitas dan diagram ini juga menggambarkan interaksi antara objek yang ada pada sistem. Hampir sama seperti *sequence* diagram, akan tetapi *communication* diagram lebih menekankan kepada peranan masing-masing objek pada sistem.



Gambar 2.14 *Communication* diagram
Sumber: (http://www.pengertianku.net)

g. *Deployment* diagram

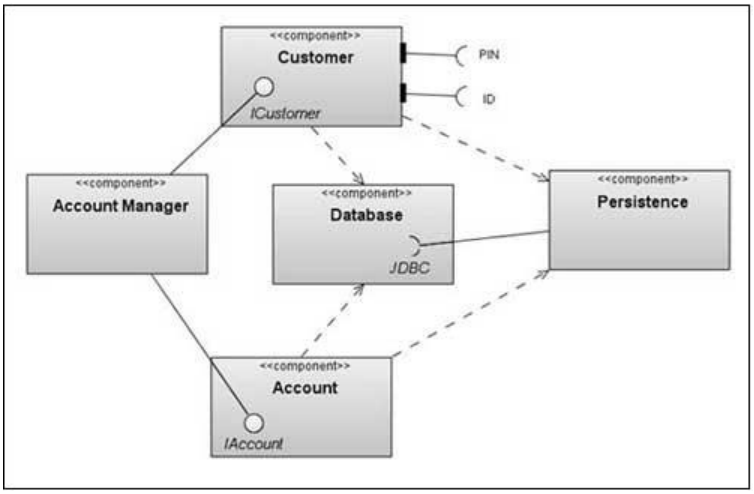
Yang menunjukkan tata letak suatu sistem secara fisik, dapat juga dikatakan untuk menampilkan bagian-bagian *software* yang terdapat pada *hardware* dan digunakan untuk menerapkan suatu sistem serta hubungan antara komponen *hardware*. Jadi, *deployment* diagram intinya untuk menunjukkan letak *software* pada *hardware* yang digunakan sistem.



Gambar 2.15 *Deployment* diagram
Sumber: (http://www.pengertianku.net)

h. *Component* diagram

Component diagram yaitu yang menggambarkan *software* pada suatu sistem. *Component* diagram merupakan penerapan *software* dari satu atau pun lebih kelas, dan biasanya berupa *file* data atau *.exe*, *source code*, tabel, dokumen, dsb.



Gambar 2.16 *Component* diagram
Sumber: (http://www.pengertianku.net)

i. *Object* diagram

Object diagram yaitu salah satu jenis diagram pada UML yang menggambarkan objek-objek pada suatu sistem dan hubungan antaranya.

j. *Composite structure* diagram

Salah satu jenis diagram pada UML yang menggambarkan struktur internal penklasifikasian (*class, component*, atau *use case*) dan termasuk titik-titik interaksi penklasifikasian ke bagian lainnya dari suatu sistem. Ini hampir mirip dengan *class* diagram, akan tetapi *composite structure* diagram menggambarkan bagian-bagian dari individu kelas saja bukan semua kelas.

k. *Interaction overview* diagram

Berguna untuk menvisualisasikan kerjasama dan hubungan antara *activity* diagram dengan *sequence* diagram.

l. *Package* diagram

Salah satu jenis diagram pada UML yang digunakan untuk mengelompokkan kelas dan juga menunjukkan bagaimana elemen model akan disusun serta menggambarkan ketergantungan antar paket-paket.

m. Diagram *timing*

Diagram *timing* yaitu salah satu diagram pada UML yang disebut sebagai bentuk lain interkasi diagram, di mana fokus utamanya kepada waktu. Diagram timing digunakan untuk menunjukkan faktor-faktor yang membatasi waktu antara perubahan *state* terhadap objek yang berbeda.

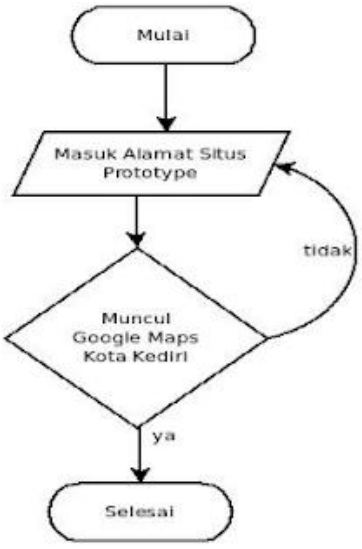
18. Metode pengujian

a. Pengujian *white box*

Pengujian yang didasarkan pada detail prosedur dan alur logika kode program. Pada kegiatan *white box testing*, tester melihat *source code program* dan menemukan *bugs* dari kode program yang diuji. Intinya *whitebox testing* adalah pengujian yang dilakukan sampai kepada detail pengecekan kode program.

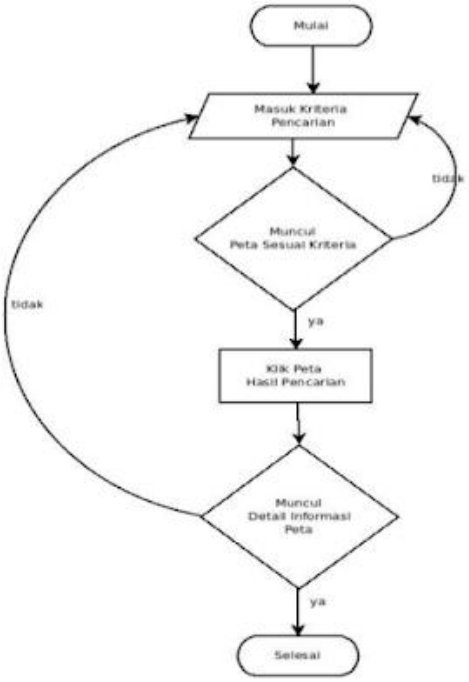
Kegiatan Tester : melihat kode program -> membuat *test case* untuk mencari kesalahan / *bugs* / *error* dari kode program yang dibuat oleh *programmer*.

Pengujian *white box* merupakan metode perancangan *test case* yang menggunakan struktur kontrol dari perancangan prosedural dalam mendapatkan *test case*. Contoh pengujian dilakukan dengan skema *flowchart diagram* gambar dibawah ini.



Gambar 2.17 *White box* pengujian pertama
Sumber: (http://www.kuliahkomputer.com)

Pengujian dilakukan untuk melihat apakah halaman awal ketika dibuka muncul peta seluruh wilayah kota. Hasil dari *flowchart* menunjukkan wilayah kota muncul saat masuk halaman awal alamat *prototype* aplikasi. Pengujian Kedua dilakukan pengujian logika apakah alamat kriteria muncul saat dimasukkan filter sesuai kriteria. Hasil dari pengujian muncul sesuai dengan kriteria yang dimasukkan. Berikut ini *flowchart* pengujian kedua pada gambar dibawah.



Gambar 2.18 *White box* pengujian kedua
Sumber: (http://www.kuliahkomputer.com)

Analisa dari hasil pengujian *white box* menunjukkan aplikasi sudah sesuai harapan dari sisi ketepatan logika yang ada pada *protoytpe*.

b. Pengujian *black box*

Pengujian yang didasarkan pada detail aplikasi seperti tampilan aplikasi, fungsi-fungsi yang ada pada aplikasi, dan kesesuaian alur fungsi

dengan bisnis proses yang diinginkan oleh *customer*. Pengujian ini tidak melihat dan menguji *source code* program. Kegiatan Tester:

1) Membuat *test case* untuk menguji fungsi-fungsi yang ada pada aplikasi.
2) Membuat *test case* untuk menguji kesesuaian alur kerja suatu fungsi di aplikasi dengan *requirement* yang dibutuhkan *customer* untuk fungsi tersebut.
3) Mencari *bugs / error* dari tampilan (*interface*) aplikasi.

Pengujian *Black Box* yang fokus pada persyaratan fungsional dari perangkat lunak yang dibangun. Berdasarkan rencana pengujian, maka dapat dilakukan pengujian *black box* pada *prototype* aplikasi peta hama dan penyakit ikan yang dijelaskan pada contoh dibawah ini.

Aktivitas Pengujian	Realisasi yang diharapkan	Hasil Pengujian	Kesimpulan
Masuk halaman awal	Terhubung dengan API Google Maps	Muncul Peta Kota Kediri.	[x] Diterima [ ] Ditolak
Klik Dropdown Filter criteria	Muncul kriteria sesuai dengan label	Dropdown dapat berfungsi sesuai yang diharapkan.	[x] Diterima [ ] Ditolak
Klik tombol "Cari Lokasi"	Muncul peta Lokasi sesuai kriteria yang dimasukkan	Tombol "Cari Lokasi" berfungsi sesuai yang diharapkan.	[x] Diterima [ ] Ditolak
Klik tombol "Cari Lokasi"	Muncul peta lokasi dengan warna yang berbeda.	Ada beberapa peta lokasi yang memiliki warna yang sama	[ ] Diterima [x] Ditolak
Klik pada peta lokasi	Muncul informasi detail lokasi pada setiap peta	Popup informasi berfungsi sesuai yang diharapkan	[x] Diterima [ ] Ditolak
Akses Melalui perangkat *smart phone* layar ≥5 inch	Tampilan sesuai dengan ukuran layar smartphone	Belum mampu menampilkan tampilan sesuai dengan layar smartphone	[ ] Diterima [x] Ditolak

Gambar 2.19 *Black box* pengujian pertama
Sumber: (http://www.kuliahkomputer.com)

Hasil dari pengujian dengan *black box* ada beberapa hasil pengujian yang belum sesuai dengan realisasi yang diharapkan sehingga dilakukan beberapa perbaikan sesuai dengan masalah yang didapatkan, berikut perbaikan yang dilakukan:

1) Perbaikan pertama dilakukan pengecekan warna peta yang ada kemunculan warna yang sama. Langkah yang dilakukan untuk memperbaiki *bug* tersebut dengan mengecek logika warna dan menambahkan *field status* warna untuk memudahkan pemberian warna sehingga warna muncul sesuai yang diharapkan.
2) Perbaikan kedua dilakukan pengecekan fungsi tampilan yang tidak sesuai dengan perbedaan pada akses *device* layar yang berbeda. Langkah yang dilakukan dengan menambahkan *style* menggunakan *bootstrap*. *Bootstrap* merupakan *framework* untuk membangun desain web secara *responsive*. Artinya, tampilan web yang dibuat oleh *bootstrap* akan menyesuaikan ukuran layar dari *browser* yang kita gunakan baik di *desktop,* tablet ataupun *mobile-device*.

Setelah perbaikan dilakukan maka dilakukan pengujian ulang dengan *black box*. Pengujian hanya dilakukan pada fungsi yang belum sesuai yang diharapkan. Pengujian ulang *black box* pada *prototype* aplikasi pemetaan hama dan penyakit ikan yang dijelaskan pada gambar di bawah ini.

Aktivitas Pengujian	Realisasi yang diharapkan	Hasil Pengujian	Kesimpulan
Klik tombol "Cari Lokasi"	Muncul peta lokasi dengan warna yang berbeda.	Warna peta pada *prototype* aplikasi berfungsi sesuai yang diharapkan	[x] Diterima [ ] Ditolak
Akses Melalui perangkat *smart phone* layar ≥5 inch	Tampilan sesuai dengan ukuran layar *smartphone*	Tampilan *prototype* aplikasi berfungsi sesuai yang diharapkan	[x] Diterima [ ] Ditolak

Gambar 2.20 *Black box* pengujian kedua
Sumber: (http://www.kuliahkomputer.com)

19. *Flowchart* (bagan alir)

Flowchart adalah penggambaran secara grafik dari langkah-langkah dan urutan-urutan prosedur dari suatu program. *Flowchart* menolong analis dan *programmer* untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian.

Flowchart biasanya mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut.

Ada beberapa jenis - jenis *flowchart* diantaranya:

a. *System flowchart*

System flowchart dapat didefinisikan sebagai bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Bagan ini menjelaskan urut-urutan dari prosedur-prosedur yang ada di dalam sistem. Bagan alir sistem menunjukkan apa yang dikerjakan di sistem.

b. *Document flowchart*

Bagan alir dokumen (*document flowchart*) atau disebut juga bagan alir formulir (*form flowchart*) atau *paperwork flowchart* merupakan bagan

alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya.

c. *Schematic flowchart*

Bagan alir skematik (*schematic flowchart*) merupakan bagan alir yang mirip dengan bagan alir sistem, yaitu untuk menggambarkan prosedur di dalam sistem. Perbedaannya adalah bagan alir skematik selain menggunakan simbol-simbol bagan alir sistem, juga menggunakan gambar-gambar komputer dan peralatan lainnya yang digunakan. Maksud penggunaan gambar-gambar ini adalah untuk memudahkan komunikasi kepada orang yang kurang paham dengan simbol-simbol bagan alir. Penggunaan gambar-gambar ini memudahkan untuk dipahami, tetapi sulit dan lama menggambarnya.

d. Program *flowchart*

Bagan alir program (*program flowchart*) merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program. Bagan alir program dibuat dari derivasi bagan alir sistem. Bagan alir program dapat terdiri dari dua macam, yaitu bagan alir logika program (*program logic flowchart*) dan bagan alir program komputer terinci (*detailed computer program flowchart*). Bagan alir logika program digunakan untuk menggambarkan tiap-tiap langkah di dalam program komputer secara logika. Bagan alat-logika program ini dipersiapkan oleh analis sistem. Bagan alir program komputer terinci (*detailed computer program flow-*

chart) digunakan untuk menggambarkan instruksi-instruksi program komputer secara terinci. Bagan alir ini dipersiapkan oleh pemrogram.

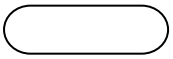
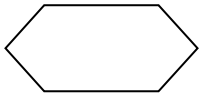
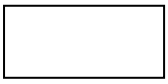
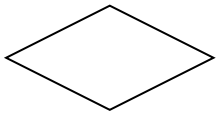
e. *Process flowchart*

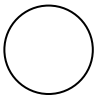
Bagan alir proses (*process flowchart*) merupakan bagan alir yang banyak digunakan di teknik industri. Bagan alir ini juga berguna bagi analis sistem untuk menggambarkan proses dalam suatu prosedur.

Bagan alir juga terdiri dari simbol-simbol yang mewakili fungsi-fungsi langkah program yang ada dan garis alir (*flowlines*) menunjukkan urutan simbol-simbol yang akan dikerjakan.

Adapun Simbol-simbol Bagan Alir (*flowchart*) menurut ANSI (*American Natioanal Standard Institute*) yakni:

Tabel 2.6 Simbol-simbol bagan alir (*flowchart*)

No.	Simbol-simbol	Keterangan
1.		Terminal symbol (*terminal symbol*), digunakan untuk menunjukkan awal dan akhir dari program
2.		Simbol persiapan (*Preparation symbol*), digunakan untuk memberikan nilai awal pada suatu variable
3.		Simbol pengolahan (*processing symbol*) digunakan untuk pengolahan arthmatika dan pemindahan data.
4.		Simbol keputusan (*Decision symbol*), digunakan untuk mewakili operasi perbandingan logika.

No.	Simbol-simbol	Keterangan
5.		Simbol proses terdefinsi (*predefined process symbol*) digunakan untuk proses yang detailnya dijelaskan terpisah, misalnya dalam bentuk subroutne.
6.		Simbol penghubung (*Connector symbol*), digunakan untuk menunjukkan hubungan arus proses yang terputus masih dalam halaman yang sama.

20. Kerangka berpikir

Untuk memperjelas alur penelitian ini dibuatkan kerangka berpikir sebagai berikut:

Internet of Things (IoT) merupakan suatu konsep mengenai benda dalam kehidupan sehari-hari dengan sensor, *software*, atau teknologi lain di dalamnya, memungkinkannya terhubung dengan perangkat serta sistem lain melalui jaringan internet.

↓

Implementasi IoT sudah banyak diterapkan di berbagai macam bidang. Awalnya, mesin dibuat hanya untuk membantu manusia dan dioperasikan secara manual. Namun, seiring perkembangan teknologi yang begitu pesat, mesin bisa berjalan sendiri atau otomatis, tetapi dalam pemanfaatannya mesin menemui kendala jika sudah menyangkut jarak dan waktu. Dengan jarak yang begitu jauh maka mesin tidak akan bisa berinteraksi dengan mesin lain, maka implementasi IoT diterapkan pada sebuah mesin dengan pengenal IP *address* yang dapat menggunakan jaringan internet sebagai media komunikasi saling bertukar data.

↓

Implementasi IoT dengan sistem *monitoring* informasi yang dapat dilakukan di mana saja dan kapan saja selama terkoneksi dengan jaringan internet yang menggunakan 4 (sensor) pada pengolahan datanya. Serta akan dilakukan pengujian kecepatan pengiriman data sensor ke *database* hingga ditampilkan pada situs web.

↓

Diharapkan dalam penelitian ini menghasilkan sebuah sistem “Implementasi IoT sebagai alat pengolah multi sensor”.

B. Kajian hasil penelitian terdahulu

Penelitian merupakan proses pemecahan suatu masalah dengan melakukan suatu pendekatan dengan metode ilmiah untuk menyelesaikan permasalahan yang ada secara sistematis. Ada pun penelitian-penelitian sebelumnya yang memberikan penulis referensi dengan keterkaitan latar belakang masalah pada tugas akhir ini, yaitu:

1. Trisiani Dewi Hendrawati, Adnan Rafi Al Tahtawi dan Farhan Fadilah, Program Studi Teknik Komputer, Politeknik Sukabumi, Jurusan Teknik Elektro, Politeknik Negeri Bandung, tahun 2019. Melakukan penelitian dengan judul "Sistem *Monitoring* Pencemaran Air Sungai Berbasis Teknologi Sensor Nirkabel dan *Internet-of-Things*". Sistem pada penelitian ini dapat melakukan pengukuran parameter pH, kekeruhan, dan suhu air sungai dengan memanfaatkan *node* sensor dan *node* utama yang ditempatkan terpisah. Data pengukuran juga dapat ditampilkan dalam antarmuka situs web dan aplikasi android sehingga pengguna dapat melakukan akses dengan mudah selama terdapat jaringan internet.
2. Anifatul Faricha, Dimas Adiputra, Isa Hafidz, Laura Khaula Amifia dan Moch. Iskandar Riansyah, Program Studi Teknik Elektro, Fakultas Teknik Elektro, Institut Teknologi Telkom Surabaya, tahun 2019. Melakukan penelitian dengan judul "Analisa Studi tentang Perancangan Alat *Monitoring* Kualitas Air PDAM Berbasis *Internet of Things"*. Pada penelitian ini, melalui analisa studi dari berbagai sumber, pengimplementasian IoT untuk *monitoring* kualitas air PDAM di Surabaya akan dilakukan di beberapa titik. Pada pengakuisisian data

fisik beberapa sensor akan digunakan dan diintegrasikan menjadi sensor *array*. Data kualitas air yang terkumpul akan ditransmisikan ke Arduino Mega 2560 R3 yang *compatible* dengan modul *wifi* ES8266 dengan sistem *blynk cloud*.

3. Ahmad Sabiq dan Prabowo Nugroho Budisejati, Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas YARSI, tahun 2017. Melakukan penelitian dengan judul "Sistem Pemantauan Kadar pH, Suhu dan Warna pada Air Sungai Melalui Web Berbasis *Wireless* Sensor *Network*". Pada penelitian ini sensor pada setiap node dihubungkan ke arduino sebagai unit pemproses, data yang dibaca dari sensor dikirimkan ke *sink node* melalui perangkat *Xbee* nirkabel. Pada *sink node* digunakan PC yang berfungsi juga sebagai *database server* dan *web server*. Hasil dengan pengujian dengan dua penyebaran yang berbeda didapatkan hasil bahwa pembacaan sensor dapat dibaca oleh seluruh *sensor node* dan diterima oleh *sink node* serta dapat ditampilkan melalui halaman situs web yang telah dibangun.

Pada penelitian yang akan penulis buat dengan judul "Implementasi IoT sebagai alat pengolah multi sensor" ini akan memakai 4 (empat) sensor yang akan mengirimkan data ke *database* hingga ditampilkan pada situs web yang dalam pengiriman data sensornya akan diuji berapa kecepatan dalam pengiriman datanya tersebut.

BAB III

METODE PENELITIAN

A. Jenis penelitian

1. *Field research* (penelitian lapangan)

   Penulis langsung melakukan penelitian pada objek. Data dan keterangan yang diperoleh dikumpulkan melalui pengamatan maupun tanya jawab langsung sesuai dengan data dan keterangan yang dibutuhkan. Dalam hal ini:

   a. Pengamatan (*Observation*), yaitu melakukan pengamatan secara langsung di lokasi penelitian. Dari hasil observasi ini, penulis dapat mengetahui masalah-masalah yang sering terjadi serta kelemahan-kelemahan yang ada pada mekanisme yang sedang berjalan.
   b. Wawancara (*Interview*), yaitu melakukan tanya jawab secara langsung dengan berbagai pihak yang berkompeten untuk mengetahui permasalahan penelitian yang diangkat.

2. *Literature study* (studi kepustakaan)

   Melakukan pengumpulan data dengan cara mencari hasil penelitian sejenis. Dalam pencarian hasil topik penelitian sejenis, penulis akan mengumpulkan informasi sebanyak-banyaknya dari kepustakaan yang terkait. Sumber-sumber kepustakaan dapat diperoleh dari jurnal, buku, tesis, dan sumber-sumber lainnya.

B. Lokasi dan waktu

1. Lokasi penelitian

   Lokasi pelaksanaan penelitian ini dilakukan di Laboratorium IoT Gedung F Universitas Muhammadiyah Parepare, jalan Jend. Ahmad Yani KM. 6 Kota Parepare.

2. Waktu penelitian

   Waktu yang dibutuhkan untuk pelaksanaan penelitian ini adalah ± 2 (dua) bulan.

C. Alat dan bahan

Untuk melakukan penelitian ini digunakan beberapa alat dan bahan, yang terdiri dari:

1. Alat

   Alat yang digunakan, yakni:

   a. Laptop Acer dengan spesifikasi:
      1) *Processor* AMD Ryzen 5 5500U *hexa core* (*up to* 8 MB L2 cac*he*, *up to* 4.0 GHz)
      2) RAM 16 GB LPDDR4X *Memory*
      3) 512 GB SSD
      4) LCD monitor 14”
   b. NodeMCU ESP32
   c. Sensor warna TCS230
   d. *Turbidity* sensor DFRobot
   e. Sensor suhu DS18B20

f. Sensor pH 4502C

2. Bahan

Bahan penelitian yang digunakan berupa *software*, yakni:

a. Windows 11 Home

b. *Firebase realtime database*

c. Arduino IDE

d. Situs Web

D. Teknik pengumpulan data

Untuk memperoleh data-data yang dibutuhkan dalam penelitian, penulis menggunakan dua cara, yakni:

1. Secara langsung

Dalam memperoleh data-data yang dibutuhkan, penulis melakukan pengumpulan data secara langsung pada lokasi penelitian.

2. Secara tidak langsung

Mengumpulkan data hasil penelitian sejenis yang menerapkan konsep implementasi IoT sebagai alat pengolah multi sensor.

E. Teknik analisis data

Dalam Teknik analisis data yang dimaksud dalam penelitian ini ada beberapa tahapan yakni:

1. Persiapan penelitian

Pada tahapan ini penulis melakukan persiapan penelitian. Persiapan penelitian yang dimaksud adalah menyiapkan buku-buku, artikel-artikel tentang topik penelitian serta *software* yang digunakan selama penelitian.

2. Pengumpulan data

Pada tahap ini dilakukan observasi dengan peninjauan, pencatatan dan pengamatan langsung di lokasi penelitian.

3. Analisis

Pada tahap analisis, melakukan analisa terhadap sistem yang diterapkan sekarang, kemudian merumuskan masalah yang menjadi pokok penelitian sehingga dapat dibuat alternatif pemecahan masalah.

4. Perancangan

Kemudian merancang aplikasi yang ingin dibuat berdasarkan alternatif pemecahan masalah.

5. Pengujian

Setelah melakukan perancangan, dilakukan pengujian hasil perancangan yang telah dibuat. Jika hasil perancangan terdapat kekurangan atau kelemahan, maka kembali ke tahap analisis.

6. Implementasi

Setelah pada perancangan tidak terdapat kekurangan, maka sistem siap untuk digunakan oleh *user*.

F. Metode pengujian

Metode pengujian yang dilakukan pada penelitian ini menggunakan dua metode pengujian, yakni:

1. Metode *black box*

Metode pengujian *black box* untuk mengetahui berfungsinya semua fungsi pada sistem yang telah dibuat.

2. Metode *white box*

Metode pengujian *white-box* untuk menguji prosedur-prosedur yang ada.

G. Jenis data

Jenis data yang dikumpulkan dalam penelitian ini adalah data primer dan sekunder yang relevan dengan masalah yang akan diangkat.

1. Data primer

Data primer adalah data yang berasal atau data yang diperoleh langsung dari sumber-sumber yang berada di lapangan.

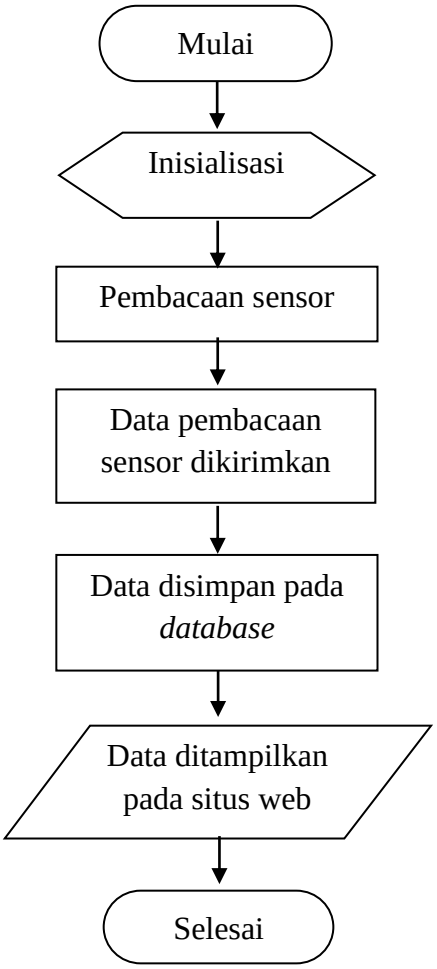
2. Data sekunder

Adalah data yang diperoleh tidak secara langsung pada objek penelitian. Peneliti mendapatkan data yang sudah jadi yang telah dikumpulkan oleh pihak lain dengan berbagai cara atau metode baik secara komersial maupun non-komersial.

H. Desain sistem

1. Sistem yang berjalan

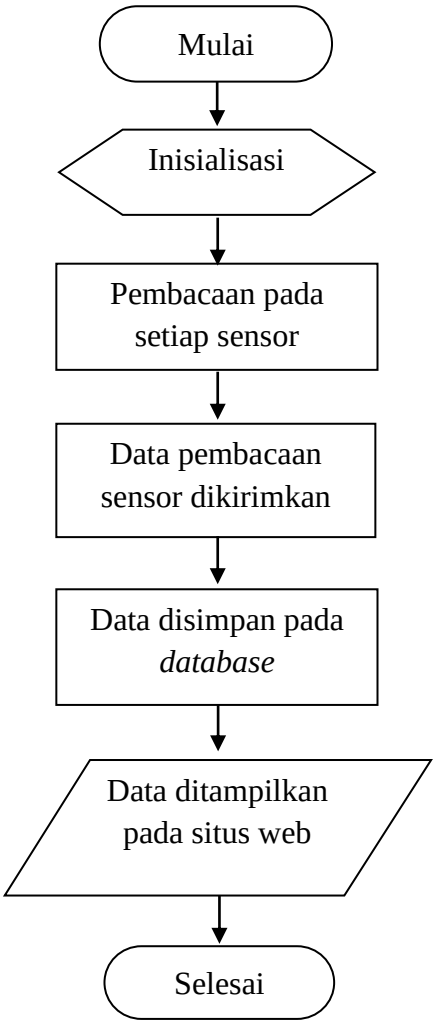
Pada sistem yang berjalan dijelaskan bahwa pertama perlu dilakukan inisialisasi, kemudian akan diproses dengan pembacaan sensor (hanya satu sensor), kemudian data akan di*input* pada *database* dan akan ditampilkan di situs web yang siap untuk memberikan informasi kepada pengguna.



Gambar 3.1 *Flowchart* sistem yang berjalan

2. Sistem yang diusulkan

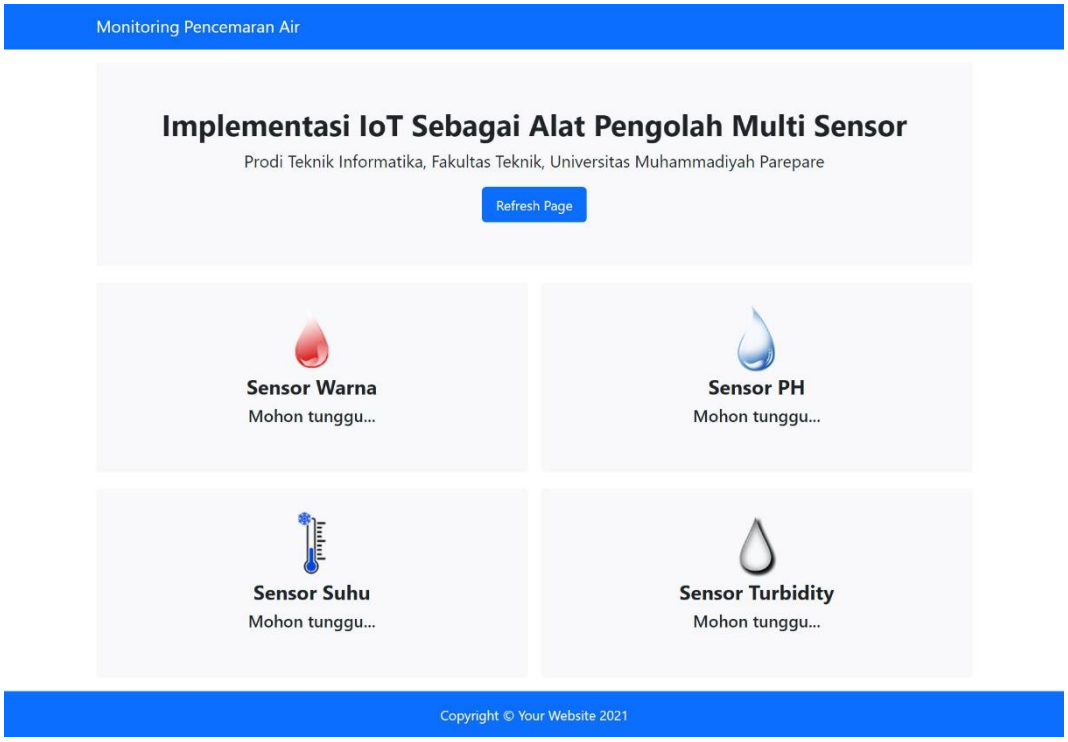
Pada desain *flowchart* di gambar 3.2 dijelaskan bahwa pertama perlu dilakukan inisialisasi, kemudian akan diproses dengan pembacaan oleh setiap sensor (kekeruhan, kadar pH, warna dan suhu air) atau multi sensor, terus berlanjut pada pengiriman data yang akan di*input* pada *database* dan akan ditampilkan di situs web yang siap untuk dipantau oleh *user*.



Gambar 3.2 *Flowchart* sistem yang diusulkan

3. Desain *user interface*

Pada gambar 3.3 di bawah, desain tersebut merupakan tampilan yang akan ditampilkan di situs web, di mana akan ditampilkan mengenai data pembacaan sensor (kekeruhan, kadar pH, suhu, dan warna) dari air yang dipantau.



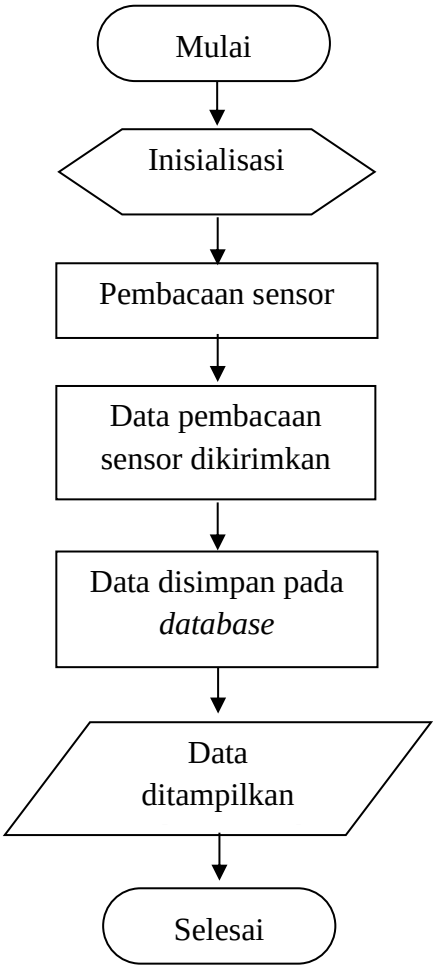
Gambar 3.3 Desain *user interface*

BAB IV

HASIL DAN PEMBAHASAN

A. Rancangan sistem

1. Sistem yang berjalan

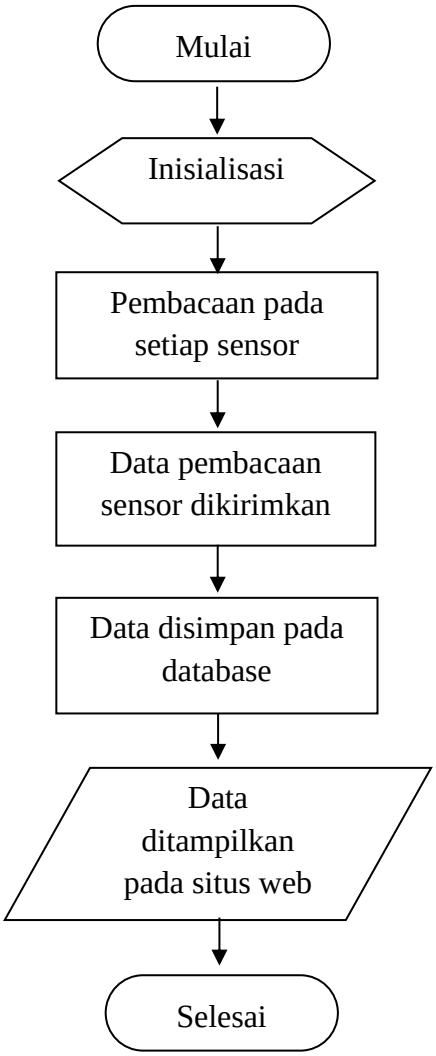


Gambar 4.1 Sistem yang berjalan

Pada gambar di atas dijelaskan dijelaskan bahwa pertama perlu dilakukan inisialisasi, kemudian akan diproses dengan pembacaan sensor

(hanya satu sensor), kemudian data akan di*input* pada *database* dan akan ditampilkan.

2. Sistem yang diusulkan



Gambar 4.2 Sistem yang diusulkan

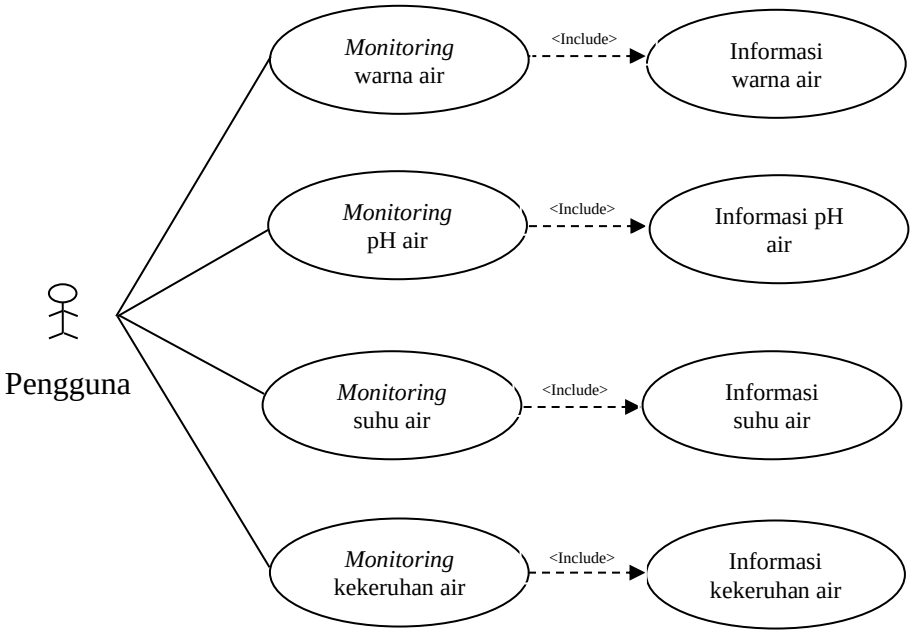
Dijelaskan pada gambar di atas bahwa pertama perlu dilakukan inisialisasi, kemudian akan diproses dengan pembacaan oleh setiap sensor (kekeruhan, kadar pH, warna dan suhu air) atau multi sensor, terus berlanjut

pada pengiriman data yang akan di*input* pada *database* dan akan ditampilkan di situs web yang siap untuk dipantau oleh *user*.

Pemodelan dalam pengembangan sistem informasi menggunakan UML. UML merupakan suatu metode pemodelan secara visual untuk sarana perancangan sistem berorientasi objek. Tipe UML yang digunakan dalam pengembangan sistem informasi adalah *use case.* Diagram dibentuk sebagai berikut:

a. *Use case*

Use case dibentuk sebagai langkah awal perancangan sistem yang menggambarkan interaksi antara sistem dan pengguna. *Use case* juga dapat mendeskripsikan tipe interaksi pengguna sistem dengan sistemnya. Berikut ini gambaran *use case* :

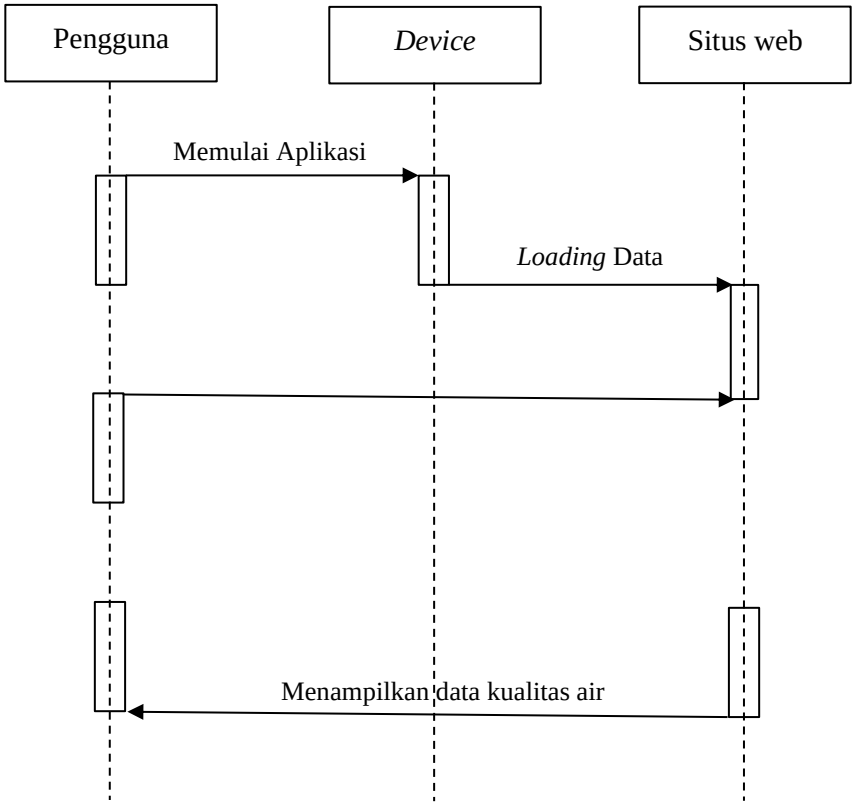


Gambar 4.3 *Use case*

Gambar di atas merupakan perancangan sistem yang dibuat menggunakan diagram *use case*. Dalam diagram *use case* ini pengguna mampu *monitoring* warna air, nilai pH air, nilai suhu air, dan *monitoring* nilai kekeruhan air.

b. *Sequence* diagram

Sequence Diagram merupakan aliran antara objek yang membentuk proses, berikut adalah *sequence* diagram :

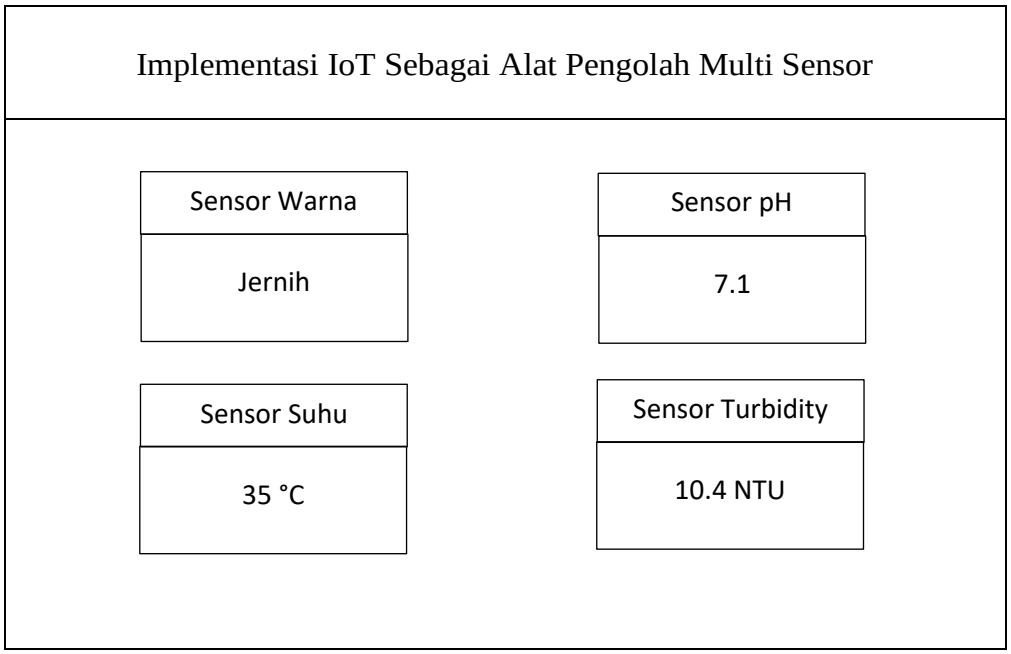


Gambar 4.4 *Sequence* diagram

3. Desain sistem

Pada proses perancangan desain aplikasi, tampilan perlu adanya sketsa desain yang digunakan untuk menggambarkan pembuatan media. Sketsa tersebut dibentuk dalam sebuah gambar. Tahap perancangan

dimulai dengan membuat sketsa aplikasi agar pembuatan program aplikasi dapat dilakukan secara terurut. Berikut rancangan *interface* dari sistem ini:



Gambar 4.5 Rancangan desain *user interface*

Pada halaman situs web ini nantinya merupakan halaman yang berisi tampilan data sensor warna, data nilai pH, data nilai suhu dan data kekeruhan air.

4. Perangkat keras (*hardware*)

Pada perancangan perangkat keras implementasi IoT sebagai alat pengolah multi sensor, mengunakan beberapa komponen perangkat keras. Berikut akan dijelaskan komponen utama yang digunakan dalam perancangan beserta rangkaiannya:

a. NodeMCU Esp32

Node MCU ESP32 adalah mikrokontroler yang dikenalkan oleh *Espressif System* yang merupakan penerus dari mikrokontroler ESP8266.

Pada mikrokontroler ini sudah tersedia modul *wifi* dalam *chip* sehingga sangat mendukung untuk membuat sistem aplikasi *Internet of Things*.

b. *Turbidity* sensor DFRobot

Turbidity sensor merupakan sensor yang berfungsi mengukur kualitas air dengan mendeteksi tingkat kekeruhannya. Sensor ini mendukung dua mode *output*, digital dan analog sehingga dapat dengan mudah diakses melalui arduino atau mikrokontroller lainnya.

Tabel 4.1 Penggunaan pin *turbidity* sensor DFRobot

No.	Pin sensor *turbidity* DFRobot	Keterangan
1.	VCC	Dihubungkan ke pin 5 volt Esp32
2.	GND	Dihubungkan ke pin GND Esp32
3.	*Out*	Dihubungkan ke pin 34 Esp32

c. Sensor pH 4502C

Merupakan kit sensor pengukur tingkat keasaman pada cairan (pH) yang menggunakan elektroda berstandar industri sebagai komponen penginderaan utamanya.

Tabel 4.2 Penggunaan pin sensor pH 4502C

No.	Pin sensor pH 4502C	Keterangan
1.	VCC	Dihubungkan ke pin 3.3 volt Esp32
2.	GND	Dihubungkan ke pin GND Esp32
3.	*Out*	Dihubungkan ke pin 33 Esp32

d. Sensor warna TCS230

Sensor warna TCS230 adalah sensor warna yang sering digunakan pada aplikasi mikrokontroler untuk pendeteksian suatu objek benda atau warna dari objek yang dipantau. Pada dasarnya sensor warna TCS230 adalah rangkaian *photodioda* yang disusun secara *matrik array* 8x8 dengan 16 buah konfigurasi *photodioda* yang berfungsi sebagai filter warna merah, 16 *photodioda* sebagai filter warna biru dan 16 *photodioda* lagi tanpa filter warna. Sensor warna TCS230 merupakan sensor yang dikemas dalam chip DIP 8 pin dengan bagian muka transparan sebagai tempat menerima intensitas cahaya yang berwarna.

Tabel 4.3 Penggunaan pin sensor warna TCS230

No.	Pin sensor warna TCS230	Keterangan
1.	VCC	Dihubungkan ke pin 3.3 volt Esp32
2.	GND	Dihubungkan ke pin GND Esp32
3.	S0	Dihubungkan ke pin 15 Esp32
4.	S1	Dihubungkan ke pin 2 Esp32
5.	S2	Dihubungkan ke pin 5 Esp32
6.	S3	Dihubungkan ke pin 4 Esp32
7.	*Out*	Dihubungkan ke pin 18 Esp32

e. Sensor suhu DS18B20

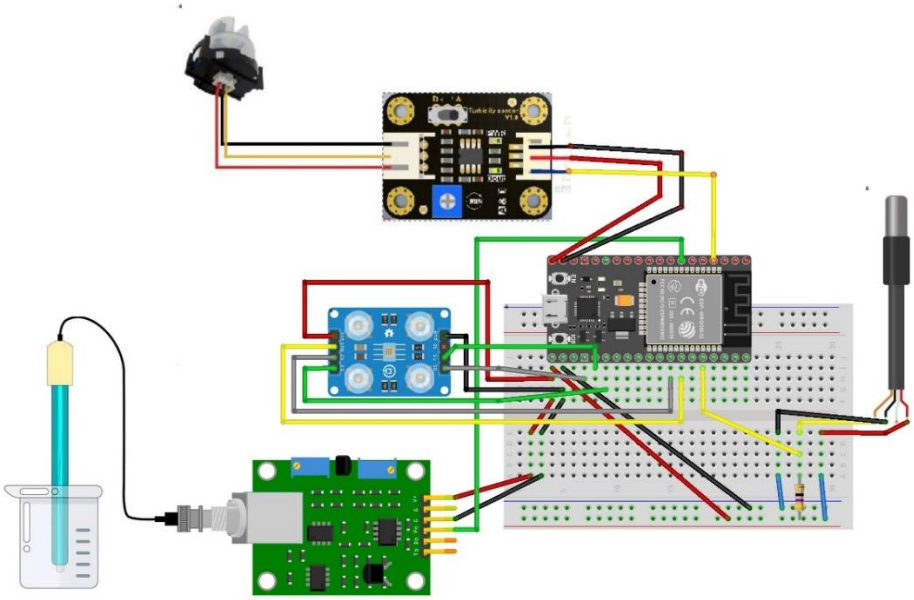
Sensor suhu DS18B20 adalah sensor suhu yang menggunakan *interface one wire*, sehingga hanya menggunakan kabel yang sedikit dalam

instalasinya. Sensor ini memiliki tipe *waterproof*, sehingga sensor ini bisa digunakan sebagai alat ukur air maupun kontrol pemanas air.

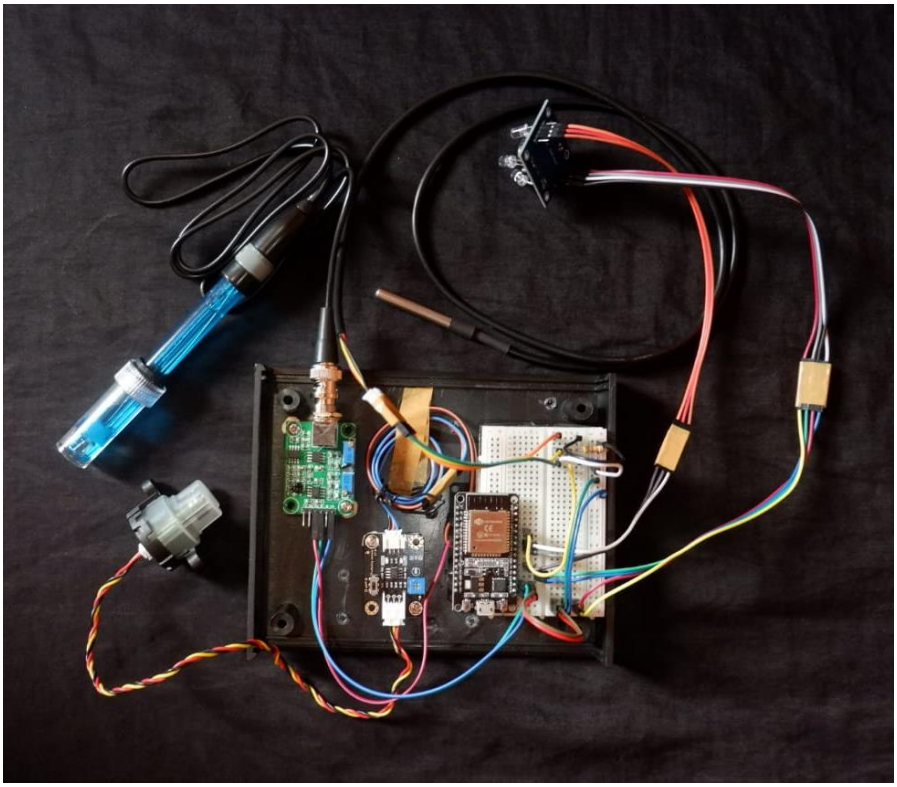
Tabel 4.4 Penggunaan pin sensor suhu DS18B20

No.	Pin sensor suhu DS18B20	Keterangan
1.	VCC	Dihubungkan ke pin 3.3 volt Esp32
2.	GND	Dihubungkan ke pin GND Esp32
3.	*Out*	Dihubungkan ke pin 21 Esp32

Adapun alur pengkabelan atau rangkaian perancangan implementasi IoT sebagai alat pengolah multi sensor bisa kita lihat pada gambar di bawah:



Gambar 4.6 Rangkaian komponen keseluruhan



Gambar 4.7 Bentuk rangkaian komponen keseluruhan

Pada gambar di atas yakni rancangan rangkaian komponen keseluruhan dapat dilihat bahwa beberapa komponen yang digunakan saling terhubung satu sama lain.

5. Perangkat lunak (*software*)

   a. Rancangan program mikrokontroler



all_hk | Arduino 1.8.15

File Edit Sketch Tools Help

all_hk §

```
#include <OneWire.h>
#include <DallasTemperature.h>
#include <WiFi.h>
#include <FirebaseESP32.h>

#define WIFI_SSID "HK"
#define WIFI_PASSWORD "helpmeguys"
#define FIREBASE_HOST "https://iotmultisensor-default-rtdb.firebaseio.com/"
#define FIREBASE_AUTH "dxQVsvUnM4Il9N9xl7J2dmyVEMz5zWAN8npZQ4GK"
FirebaseData fbdo;

//TURBIDITY Pin vcc = 5 volt biru = analog, merah = vcc, hitam = GND
int pinTurb = 34;
float V;
float kekeruhan;
```

```
//PH. Pin vcc = 3.3 volt
const int ph_Pin  = 33;
float Po = 0;
float PH_step;
int nilai_analog_PH;
double TeganganPh;
//untuk kalibrasi pH
float PH4 = 3.30;
float PH7 = 2.61;

//suhu.  Pin vcc = 3.3 volt merah= vcc  kuning= data  hitam= GND
#define ONE_WIRE_BUS 21 // 21 = pin data sensor suhu
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensorSuhu(&oneWire);
float suhu;

// sensor warna pin vcc = 3.3 volt
#define S0 15
#define S1 2
#define S2 5
#define S3 4
#define sensorOut 18
//Menyimpan frekuensi yang dibaca oleh fotodioda
int redFrequency = 0;
int greenFrequency = 0;
int blueFrequency = 0;
// Menyimpan warna sesuai keruh dan jernih
int redColor = 0;
int greenColor = 0;
int blueColor = 0;
float Merah;
float Hijau;
float Biru;

void setup () {
    Serial.begin(115200);

  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to Wi-Fi");
  while (WiFi.status() != WL_CONNECTED)
  {
    Serial.print(".");
    delay(300);
  }
  Serial.println();
  Serial.print("Connected with IP: ");
  Serial.println(WiFi.localIP());
  Serial.println();

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  Firebase.reconnectWiFi(true);
  sensorSuhu.begin();

  //mengatur output
  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);
  //mengatur sensorOut sebagai input
  pinMode(sensorOut, INPUT);
  //menyetel penskalaan frekuensi ke 20%
  digitalWrite(S0,HIGH);
  digitalWrite(S1,HIGH);
}

void loop () {

    //pembacaan sensor kekeruhan
    int val = analogRead(pinTurb);
    V = val*(5.0/1024);
    kekeruhan= 100.00-(V/20.00)*100.00;
    Serial.print("Kekeruhan :");
    Serial.print(kekeruhan);
    Serial.println(" NTU");

    //pembacaan sensor pH
    nilai_analog_PH = analogRead(ph_Pin);
    TeganganPh = 3.3 / 4095.0 * nilai_analog_PH;
    PH_step = (PH4 - PH7) / 3;
    Po = 7.00 + ((PH7 - TeganganPh) / PH_step);     //Po = 7.00 + ((teganganPh7 - TeganganPh) / PhStep);
    Serial.print("Nilai PH = ");
    Serial.println(Po, 2);
```

```
  //pembacaan sensor suhu
  sensorSuhu.requestTemperatures();
  suhu = sensorSuhu.getTempCByIndex(0);
  Serial.print("Suhu: ");
  Serial.print(suhu);
  Serial.println("°C");

  //pembacaan sensor warna
  digitalWrite(S2,LOW);
  digitalWrite(S3,LOW);
  //membaca frekuensi keluaran
  redFrequency = pulseIn(sensorOut, LOW);
  redColor = map(redFrequency, 24,111, 255,0);
  Serial.print("M = ");
  Serial.print(redColor);

  digitalWrite(S2,HIGH);
  digitalWrite(S3,HIGH);
  //membaca frekuensi keluaran
  greenFrequency = pulseIn(sensorOut, LOW);
  greenColor = map(greenFrequency, 41, 191, 255, 0);
  Serial.print(" H = ");
  Serial.print(greenColor);

  digitalWrite(S2,LOW);
  digitalWrite(S3,HIGH);
  //membaca frekuensi keluaran
  blueFrequency = pulseIn(sensorOut, LOW);
  blueColor = map(blueFrequency, 34, 156, 255, 0);
  Serial.print(" B = ");
  Serial.print(blueColor);

 //Memeriksa warna dan cetakan yang terdeteksi saat ini
 //pesan di monitor serial
 if(redColor > greenColor && redColor > blueColor){
     Serial.println(" - Keruh");
     Firebase.setString(fbdo, "/warna", "Keruh");
 }
 if(greenColor > redColor && greenColor > blueColor){
   Serial.println(" - Keruh");
   Firebase.setString(fbdo, "/warna", "Keruh");
 }
 if(blueColor > redColor && blueColor > greenColor){
   Serial.println(" - Jernih");
   Firebase.setString(fbdo, "/warna", "Jernih");
 }
       Serial.println("");
       Serial.println("Kirim data ke Firebase");
       firebaseSend ();
       Serial.println("OK Success");
       delay(500);
       }

void firebaseSend ()  {
     Firebase.setFloat(fbdo, "/suhu", suhu);
     Firebase.setFloat(fbdo, "/kekeruhan", kekeruhan);
     Firebase.setFloat(fbdo, "/pH", Po);
   }
```

Gambar 4.8 Rancangan program mikrokontroler

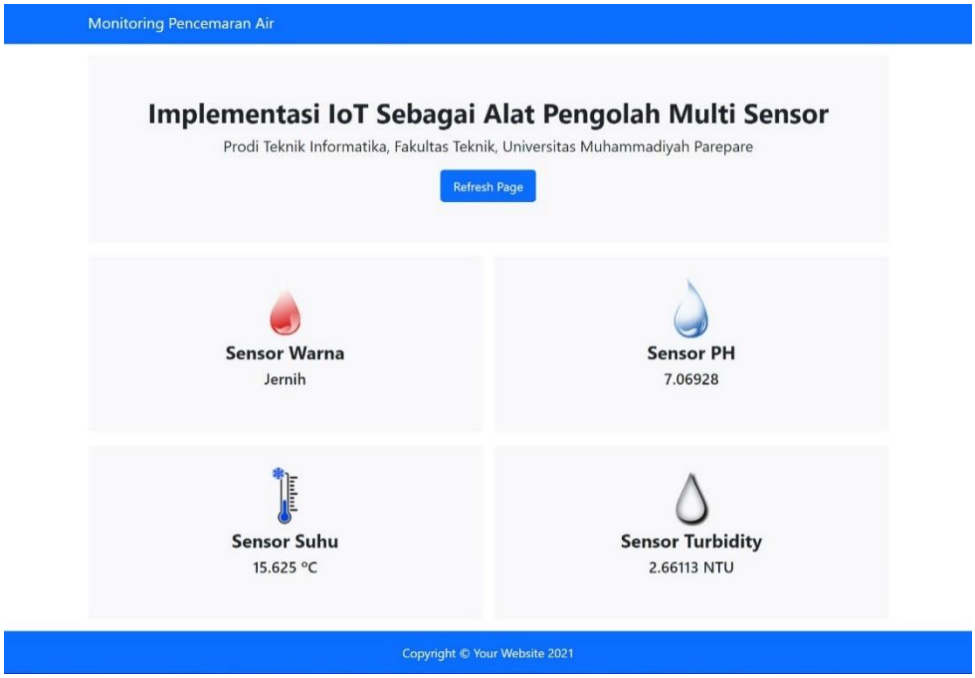
Pada gambar rancangan program mikrokontroler di atas menggunakan *arduino ide* (*sketch*) yang merupakan sebuah *software* untuk memprogram *arduino* dan komponen pendukung lainnya dengan menggunakan *syntax* pemrograman bahasa C. *Arduino ide* dibuat dari bahasa pemrograman java yang dilengkapi dengan *library* C/C++ yang membuat operasi *input/output* lebih mudah.

b. Rancangan situs web

Implementasi IoT sebagai alat pengolah multi sensor merupakan rancangan *monitoring* pencemaran atau kualitas air pada situs web dengan proses kerja secara otomatis.

1) Tampilan situs web

Tampilan ini merupakan tampilan untuk memantau informasi warna air, nilai pH air, nilai suhu air dan nilai kekeruhan air.



Gambar 4.9 Tampilan situs web

2) Tampilan sensor warna air

Tampilan ini merupakan tampilan akan kualitas air yang dipantau berupa jernih ataukah keruh.



Gambar 4.10 Tampilan kualitas warna air

3) Tampilan nilai sensor pH air

Pada tampilan di mana nilai pH dari air yang diukur kualitasnya akan ditampilkan.



Gambar 4.11 Tampilan nilai pH air

4) Tampilan nilai suhu air

Tampilan ini menampilkan nilai suhu dari air. Menampilkan berapa °C air tersebut.



Gambar 4.12 Tampilan nilai suhu air

5) Tampilan nilai kekeruhan air

Ditampilan ini menampilkan berapa NTU (*Nephelometric Turbidity Unit*) dari suatu air yang diukur. Air yang semakin keruh berarti nilai NTU semakin tinggi.



Gambar 4.13 Tampilan nilai kekeruhan air

B. Pengujian

1. Metode Pengujian

   Metode pengujian sistem merupakan proses eksekusi suatu program atau sistem keseluruhan. Pengujian menunjukan bahwa fungsi perangkat keras dan perangkat lunak bekerja sesuai dengan spesifikasi persyaratan kinerja telah terpenuhi.

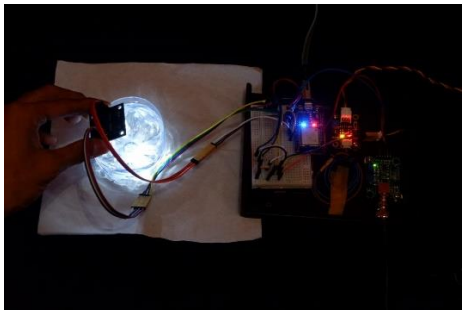
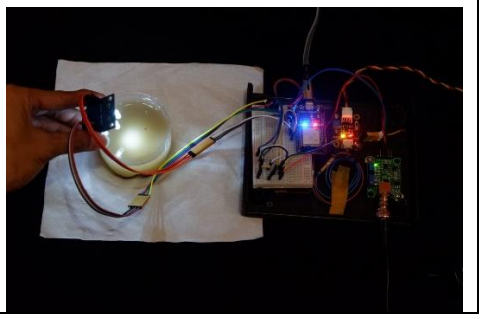
2. Pengujian *black box*

   Pengujian *black box* merupakan suatu pengujian pada *interface* perangkat lunak. Pengujian *black box* merupakan kemungkinan rekayasa perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Meskipun didesain sedemikian rupa untuk mengungkap kesalahan. Pengujian *black box* digunakan untuk memperlihatkan bahwa fungsi-fungsi perangkat lunak adalah operasional bahwa *input* diterima dengan baik dan *output* dihasilkan dengan tepat serta integritas informasi *external* (misalnya *file* data).

   Pengujian *black box* menguji beberapa aspek dasar suatu sistem dengan sedikit memperhatikan struktur logika *internal* perangkat lunak tersebut.

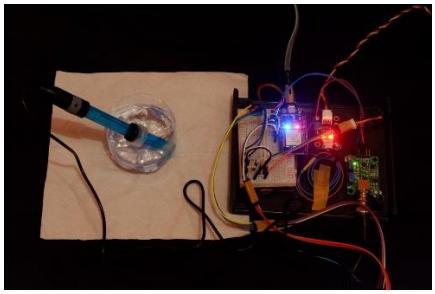
1) Pengujian sensor warna TCS230

Tabel 4.5 Pengujian sensor warna TCS230

Uji coba	Hasil	Keterangan
Jika sensor warna mendeteksi air dengan kualitas jernih, maka data yang ditampilkan pada situs web jernih	√	Berhasil. Sensor warna mendeteksi kualitas air tersebut jernih
Jika sensor warna mendeteksi air dengan kualitas keruh, maka data yang ditampilkan pada situs web keruh	√	Berhasil. Sensor warna mendeteksi kualitas air tersebut keruh
Screenshot dan foto		
		
		

2) Pengujian sensor pH 4502C

Tabel 4.6 Pengujian sensor pH 4502C

Uji coba	Hasil	Keterangan
Jika sensor pH diletakkan untuk mendeteksi kualitas air, maka data nilai pH air akan ditampilkan di situs web	√	Berhasil. Data kualitas pH air ditampilkan di situs web
Screenshot dan foto		
		

Dalam pengujian sensor pH, dilakukan juga pengujian akurasi sensor dengan melakukan perbandingan nilai pH air dari sensor pH dan pH meter digital oleh mediatech. Dalam pengujian keakuratan nilai pH, dilakukan dengan pengambilan nilai pH air yang berbeda-beda.

Tabel 4.7 Pengujian perbandingan sensor pH 4502C

No.	Pembacaan sensor pH	Pembacaan pH *meter digital*	Selisih	Galat (*error*) (%)
1.	7.6	7.8	0.2	2.6
2.	7.4	7.4	0	0
3.	4	4.1	0.1	2.4
4.	7.9	8.2	0.3	3.6
5.	6.8	6.9	0.1	1.4

Total galat (*error*) %	10
Rata-rata galat (*error*) %	2

Untuk mencari galat (error) % dari tabel pengujian perbandingan sensor pH di atas menggunakan persamaan rumus sebagai berikut:

$$\textbf{Galat(\%)} = \frac{\textbf{Pembacaan sensor – pembacaan pH digital}}{\textbf{Pembacaan pH digital}} * \textbf{100\%}$$

Dan untuk mencari rata-rata galat (error) % menggunakan persamaan rumus sebagai berikut:

$$\textbf{Rata-rata galat(\%)} = \frac{\textbf{total galat(\%)}}{\textbf{Banyak pengukuran}}$$

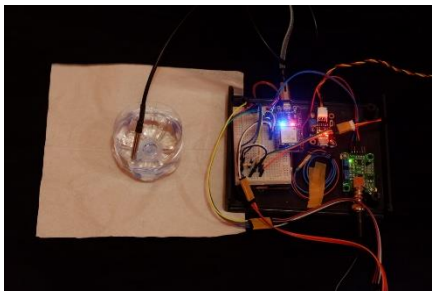
Adapun untuk akurasi menggunakan persamaan rumus sebagai berikut:

Akurasi = 100% - rata-rata galat(%) = 100% - 2% = 98%

Akurasi pada pengujian sensor pH didapatkan dengan nilai 98% dan *error* dengan nilai 2%.

3) Pengujian sensor suhu DS18B20

Tabel 4.8 Pengujian sensor suhu DS18B20

Uji coba	Hasil	Keterangan
Jika sensor suhu diletakkan untuk mendeteksi suhu air, maka data nilai °C air akan ditampilkan di situs web	√	Berhasil. Data nilai °C suhu air ditampilkan di situs web
Screenshot dan foto		
		

Pada pengujian sensor suhu, juga dilakukan pengujian akan keakuratan sensor, dengan melakukan perbandingan hasil suhu yang dibaca oleh sensor DS18B20 dan digital *thermometer* dengan melakukan pengambilan nilai suhu air pada suhu yang berbeda-beda.

Tabel 4.9 Pengujian perbandingan sensor suhu DS18B20

No.	Pembacaan sensor (°C)	Pembacaan *thermometer* (°C)	Selisih	Galat (*error*) (%)
1.	41.1	41.1	0	0
2.	30.1	30	0.1	0.3
3.	17.6	18.6	1	5.3
4.	30.8	30.8	0	0

No.	Pembacaan sensor (°C)	Pembacaan *thermometer* (°C)	Selisih	Galat (*error*) (%)
5.	22.4	22.5	0.1	0.4
			Total galat (*error*) %	6
			Rata-rata galat (*error*) %	1.2

Untuk mencari galat (error) % dari tabel 4.9 pengujian perbandingan sensor suhu menggunakan persamaan rumus sebagai berikut:

$$\textbf{Galat(\%)} = \frac{\textbf{Pembacaan sensor – pembacaan } \textbf{\textit{thermometer}}}{\textbf{Pembacaan } \textbf{\textit{thermometer}}} \textbf{ * 100\%}$$

Dan untuk mencari rata-rata galat (error) % menggunakan persamaan rumus sebagai berikut:

$$\textbf{Rata-rata galat(\%)} = \frac{\textbf{total galat(\%)}}{\textbf{Banyak pengukuran}}$$

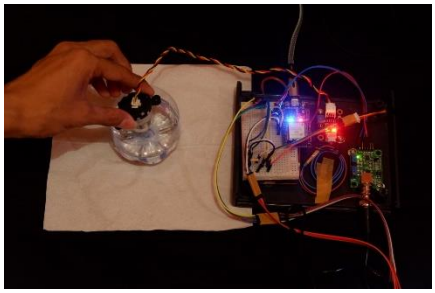
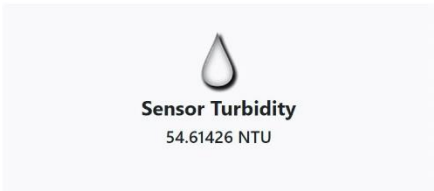
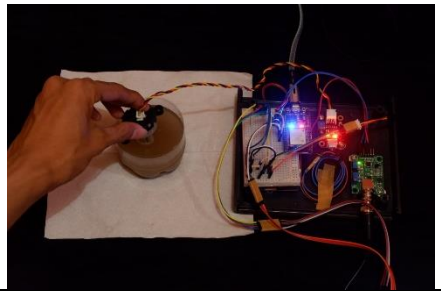
Adapun untuk mendapatkan akurasi menggunakan persamaan rumus sebagai berikut:

Akurasi = 100% - rata-rata galat(%) = 100% - 1.2% = 98.8%

Akurasi pada pengujian sensor suhu didapatkan dengan nilai 98.8% dan *error* dengan nilai 1.2%.

4) Pengujian sensor *turbidity* DFRobot

Tabel 4.10 Pengujian sensor *turbidity* DFRobot

Uji coba	Hasil	Keterangan
Jika sensor kekeruhan mendeteksi air dengan kualitas jernih, maka data NTU yang ditampilkan pada situs web rendah	√	Berhasil. Nilai NTU yang ditampilkan pada situs web rendah
Jika sensor kekeruhan mendeteksi air dengan kualitas keruh, maka data NTU yang ditampilkan pada situs web tinggi	√	Berhasil. Nilai NTU yang ditampilkan pada situs web tinggi
Screenshot dan foto		
		
		

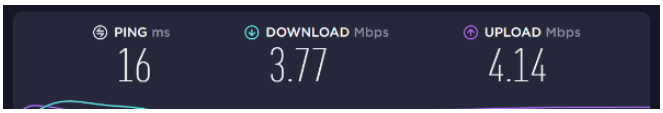
5) Pengujian kecepatan pengiriman data sensor

Pengujian kecepatan pengiriman data sensor merupakan pengujian untuk mengetahui berapa kecepatan pengiriman data dari pembacaan sensor hingga data pembacaan tersebut tampil pada situs web. Pengujian ini dilakukan dengan 3 (tiga) tahap pengujian pada

masing-masing kecepatan internet yang berbeda. Masing-masing pada kecepatan internet tersebut dilakukan 4 (empat) kali pengujian dengan menguji kecepatan pengiriman data dengan menggunakan 1 (satu) sensor, 2 (dua) sensor, 3 (tiga) sensor serta 4 (empat) atau semua sensor yang digunakan. Perhitungan kecepatan pengiriman data yang akan dilakukan berdasarkan dengan waktu *realtime* yang dapat dilihat pada *console* dalam situs web.

1) Pengujian kecepatan pengiriman data pada kecepatan pertama

Pengujian kecepatan pengiriman data pada kecepatan tahap pertama ini dilakukan pengujian dengan kecepatan internet, *download* 3.77 Mbps dan *upload* 4.14 Mbps (speedtest.net).

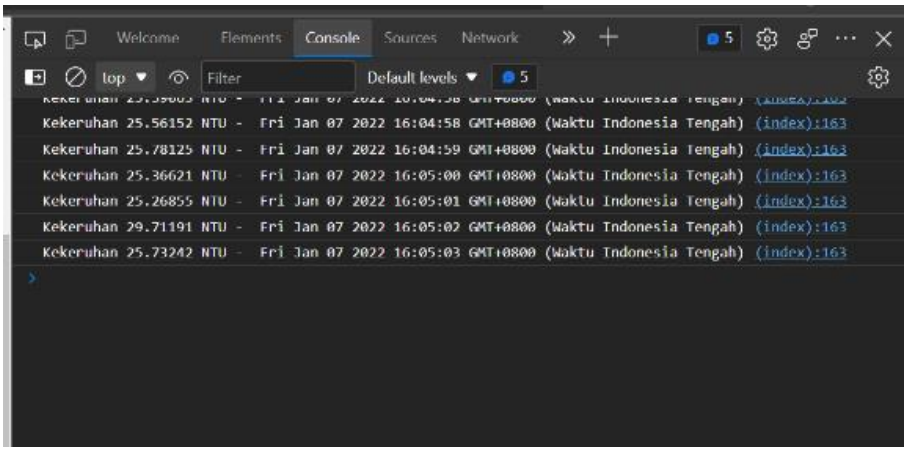


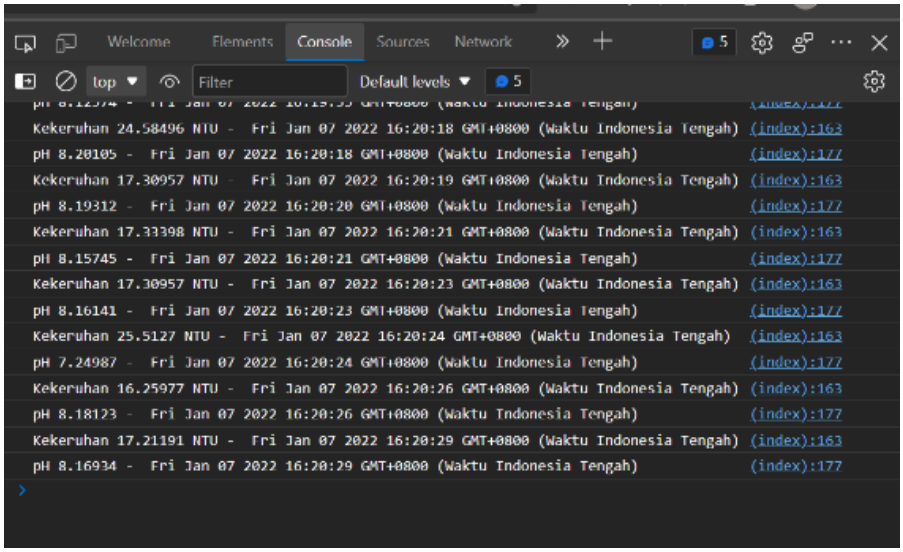
Gambar 4.14 Kecepatan internet (1)

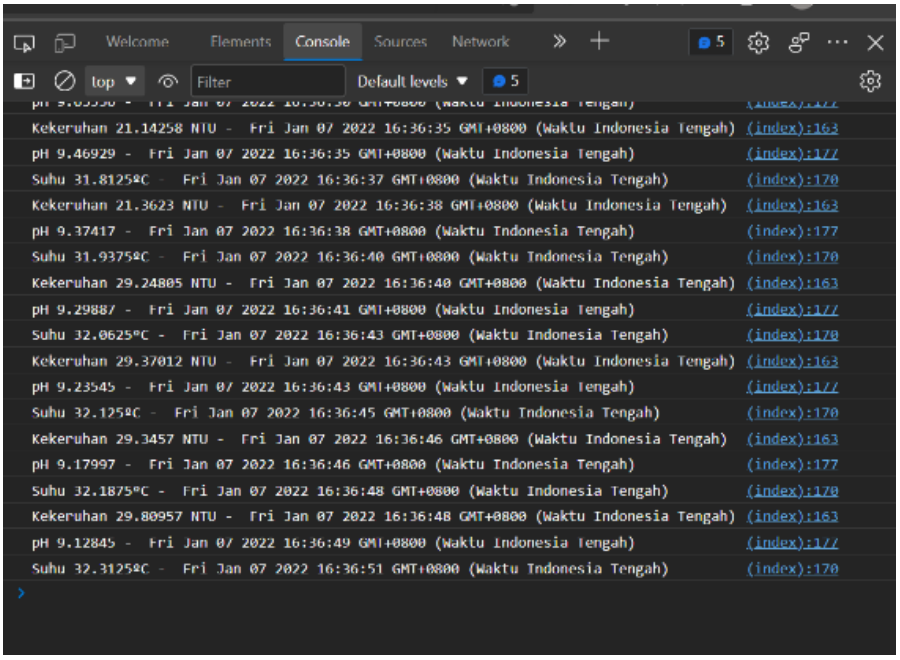
Tabel 4.11 Pengujian kecepatan pengiriman data sensor (1)

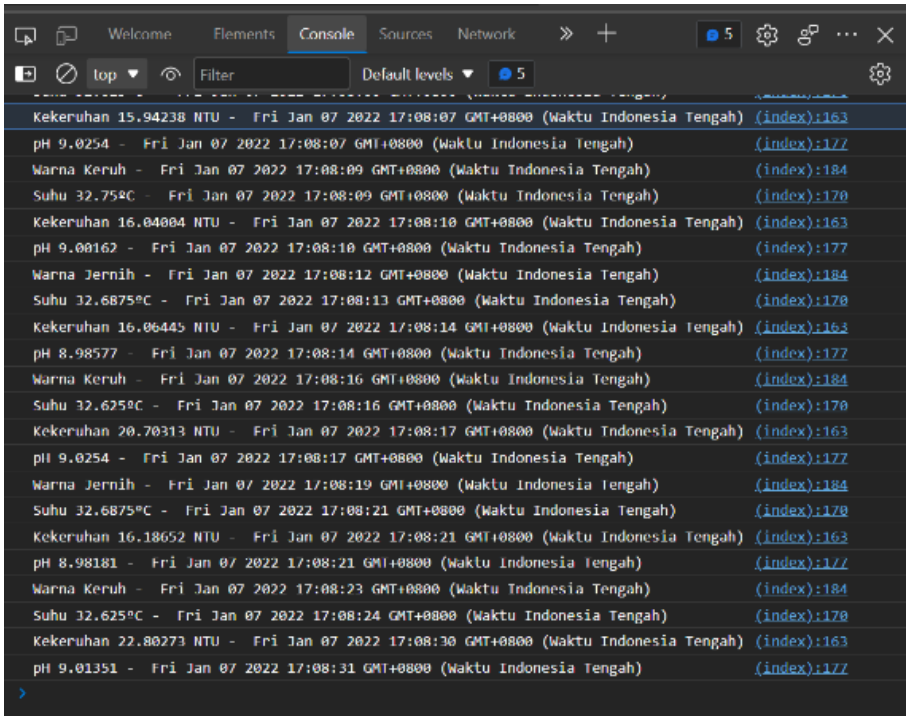
Sensor	Pengiriman data	Kecepatan
1 sensor	1	1 detik
	2	1 detik
	3	1 detik
	4	1 detik
	5	1 detik
Rata-rata		**1 detik**
2 sensor	1	2 detik
	2	1 detik
	3	2 detik
	4	1 detik

Sensor	Pengiriman data	Kecepatan
2 sensor	5	2 detik
Rata-rata		**1.6 detik**
3 sensor	1	3 detik
	2	3 detik
	3	2 detik
	4	3 detik
	5	3 detik
Rata-rata		**2.8 detik**
4 sensor	1	3 detik
	2	4 detik
	3	3 detik
	4	4 detik
	5	10 detik
Rata-rata		**4.8 detik**




Welcome
Elements
Console
Sources
Network
top
Filter
Default levels
Kekeruhan 24.58496 NTU - Fri Jan 07 2022 16:20:18 GMT+0800 (Waktu Indonesia Tengah) (index):163
pH 8.20105 - Fri Jan 07 2022 16:20:18 GMT+0800 (Waktu Indonesia Tengah) (index):177
Kekeruhan 17.30957 NTU - Fri Jan 07 2022 16:20:19 GMT+0800 (Waktu Indonesia Tengah) (index):163
pH 8.19312 - Fri Jan 07 2022 16:20:20 GMT+0800 (Waktu Indonesia Tengah) (index):177
Kekeruhan 17.33398 NTU - Fri Jan 07 2022 16:20:21 GMT+0800 (Waktu Indonesia Tengah) (index):163
pH 8.15745 - Fri Jan 07 2022 16:20:21 GMT+0800 (Waktu Indonesia Tengah) (index):177
Kekeruhan 17.30957 NTU - Fri Jan 07 2022 16:20:23 GMT+0800 (Waktu Indonesia Tengah) (index):163
pH 8.16141 - Fri Jan 07 2022 16:20:23 GMT+0800 (Waktu Indonesia Tengah) (index):177
Kekeruhan 25.5127 NTU - Fri Jan 07 2022 16:20:24 GMT+0800 (Waktu Indonesia Tengah) (index):163
pH 7.24987 - Fri Jan 07 2022 16:20:24 GMT+0800 (Waktu Indonesia Tengah) (index):177
Kekeruhan 16.25977 NTU - Fri Jan 07 2022 16:20:26 GMT+0800 (Waktu Indonesia Tengah) (index):163
pH 8.18123 - Fri Jan 07 2022 16:20:26 GMT+0800 (Waktu Indonesia Tengah) (index):177
Kekeruhan 17.21191 NTU - Fri Jan 07 2022 16:20:29 GMT+0800 (Waktu Indonesia Tengah) (index):163
pH 8.16934 - Fri Jan 07 2022 16:20:29 GMT+0800 (Waktu Indonesia Tengah) (index):177


Welcome
Elements
Console
Sources
Network
top
Filter
Default levels
Kekeruhan 21.14258 NTU - Fri Jan 07 2022 16:36:35 GMT+0800 (Waktu Indonesia Tengah) (index):163
pH 9.46929 - Fri Jan 07 2022 16:36:35 GMT+0800 (Waktu Indonesia Tengah) (index):177
Suhu 31.8125°C - Fri Jan 07 2022 16:36:37 GMT+0800 (Waktu Indonesia Tengah) (index):170
Kekeruhan 21.3623 NTU - Fri Jan 07 2022 16:36:38 GMT+0800 (Waktu Indonesia Tengah) (index):163
pH 9.37417 - Fri Jan 07 2022 16:36:38 GMT+0800 (Waktu Indonesia Tengah) (index):177
Suhu 31.9375°C - Fri Jan 07 2022 16:36:40 GMT+0800 (Waktu Indonesia Tengah) (index):170
Kekeruhan 29.24805 NTU - Fri Jan 07 2022 16:36:40 GMT+0800 (Waktu Indonesia Tengah) (index):163
pH 9.29887 - Fri Jan 07 2022 16:36:41 GMT+0800 (Waktu Indonesia Tengah) (index):177
Suhu 32.0625°C - Fri Jan 07 2022 16:36:43 GMT+0800 (Waktu Indonesia Tengah) (index):170
Kekeruhan 29.37012 NTU - Fri Jan 07 2022 16:36:43 GMT+0800 (Waktu Indonesia Tengah) (index):163
pH 9.23545 - Fri Jan 07 2022 16:36:43 GMT+0800 (Waktu Indonesia Tengah) (index):177
Suhu 32.125°C - Fri Jan 07 2022 16:36:45 GMT+0800 (Waktu Indonesia Tengah) (index):170
Kekeruhan 29.3457 NTU - Fri Jan 07 2022 16:36:46 GMT+0800 (Waktu Indonesia Tengah) (index):163
pH 9.17997 - Fri Jan 07 2022 16:36:46 GMT+0800 (Waktu Indonesia Tengah) (index):177
Suhu 32.1875°C - Fri Jan 07 2022 16:36:48 GMT+0800 (Waktu Indonesia Tengah) (index):170
Kekeruhan 29.80957 NTU - Fri Jan 07 2022 16:36:48 GMT+0800 (Waktu Indonesia Tengah) (index):163
pH 9.12845 - Fri Jan 07 2022 16:36:49 GMT+0800 (Waktu Indonesia Tengah) (index):177
Suhu 32.3125°C - Fri Jan 07 2022 16:36:51 GMT+0800 (Waktu Indonesia Tengah) (index):170

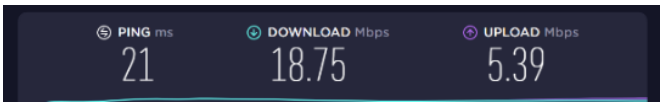


Gambar 4.15 Pengiriman data sensor (1)

Diperoleh pada pengujian kecepatan pengiriman data tahap pertama pada tabel 4.11 yakni menggunakan 1 (satu) sensor dengan nilai rata-rata 1 detik, menggunakan 2 (dua) sensor dengan nilai rata-rata 1.6 detik, 3 (tiga) sensor dengan nilai rata-rata 2.8 detik, serta menggunakan 4 (empat) sensor dengan nilai rata-rata 4.8 detik. Dengan menggunakan kecepatan internet *download* 3.77 Mbps dan *upload* 4.14 Mbps dihasilkan bahwa pengiriman data sensor akan lebih cepat dengan semakin sedikitnya sensor yang digunakan dan jika semakin banyak sensor yang digunakan, maka *delay* atau waktu pengiriman data sensor akan bertambah.

2) Pengujian kecepatan pengiriman data pada kecepatan kedua

Pengujian kecepatan pengiriman data pada kecepatan tahap kedua dilakukan pengujian dengan kecepatan internet, *download* 18.75 Mbps dan kecepatan *upload* 5.39 Mbps (speedtest.net).

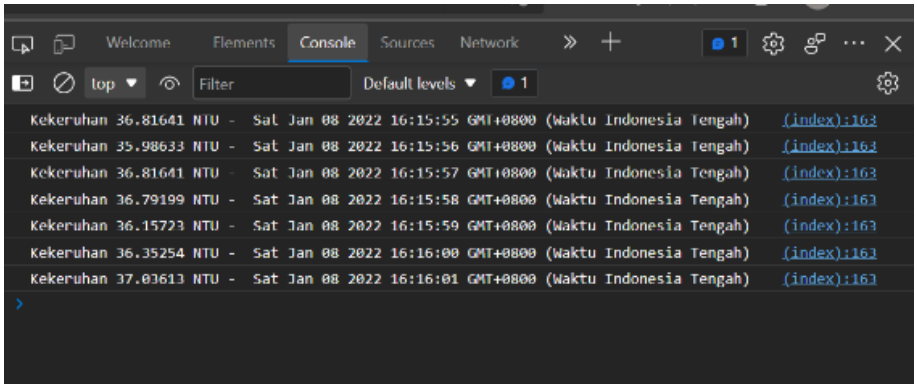


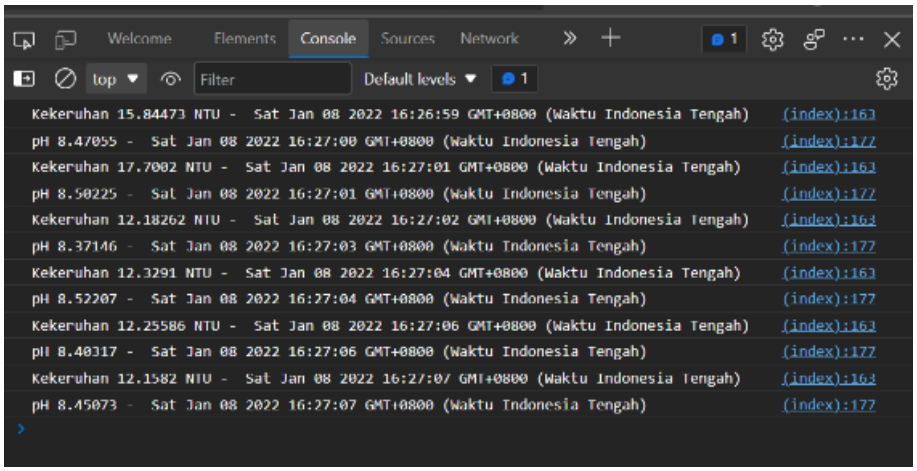
Gambar 4.16 Kecepatan internet (2)

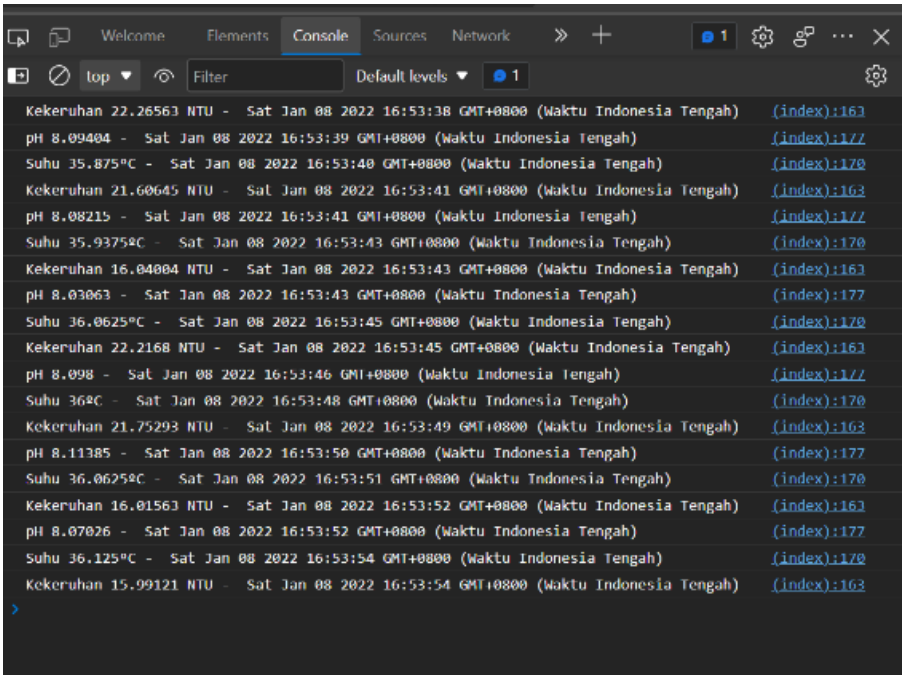
Tabel 4.12 Pengujian kecepatan pengiriman data sensor (2)

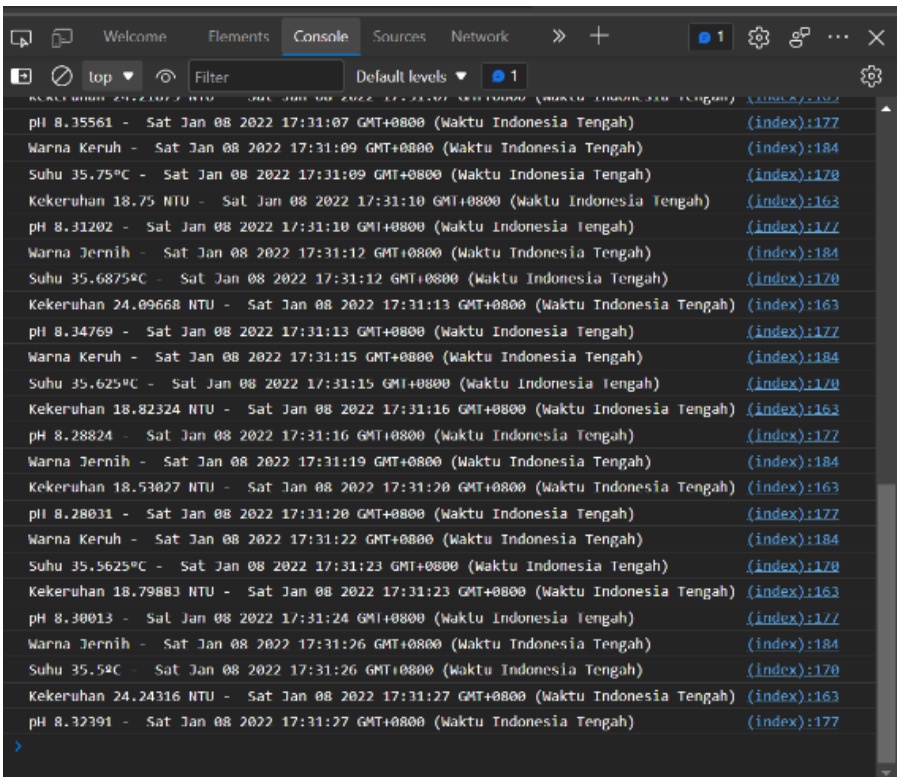
Sensor	Pengiriman data	Kecepatan
1 sensor	1	1 detik
	2	1 detik
	3	1 detik
	4	1 detik
	5	1 detik
Rata-rata		**1 detik**
2 sensor	1	1 detik
	2	2 detik
	3	1 detik
	4	2 detik
	5	1 detik
Rata-rata		**1.4 detik**
3 sensor	1	3 detik
	2	2 detik
	3	3 detik
	4	3 detik
	5	3 detik
Rata-rata		**2.8 detik**
4 sensor	1	3 detik
	2	3 detik

Sensor	Pengiriman data	Kecepatan
4 sensor	3	3 detik
	4	4 detik
	5	4 detik
Rata-rata		**3.4 detik**









Gambar 4.17 Pengiriman data sensor (2)

Hasil pada pengujian kecepatan pengiriman data tahap kedua dapat dilihat pada tabel 4.12, menggunakan 1 (satu) sensor dengan

nilai kecepatan rata-rata 1 detik, 2 (dua) sensor dengan nilai rata-rata 1.4 detik, 3 (tiga) sensor dengan nilai rata-rata 2.8 detik, dan 4 (sensor) dengan nilai rata-rata 3.4 detik. Sama seperti pengujian tahap pertama, semakin sedikit sensor yang digunakan maka semakin cepat pengiriman data dilakukan.

3) Pengujian kecepatan pengiriman data pada kecepatan ketiga

Pengujian kecepatan pengiriman data pada kecepatan tahap ketiga dilakukan pengujian dengan kecepatan internet, *download* 41.82 Mbps dan *upload* 2.08 Mbps (speedtest.net).

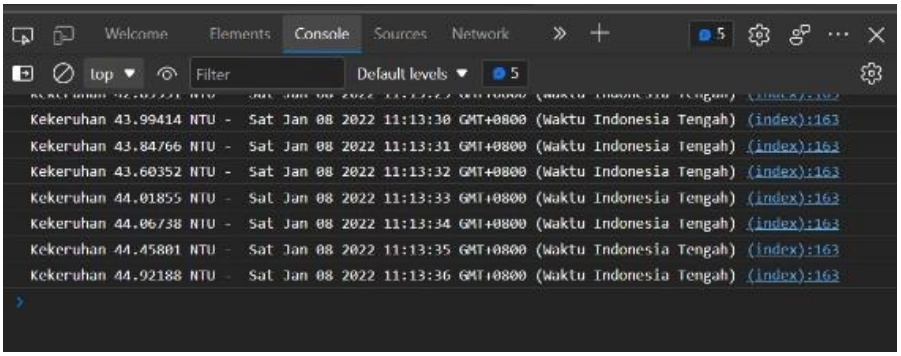


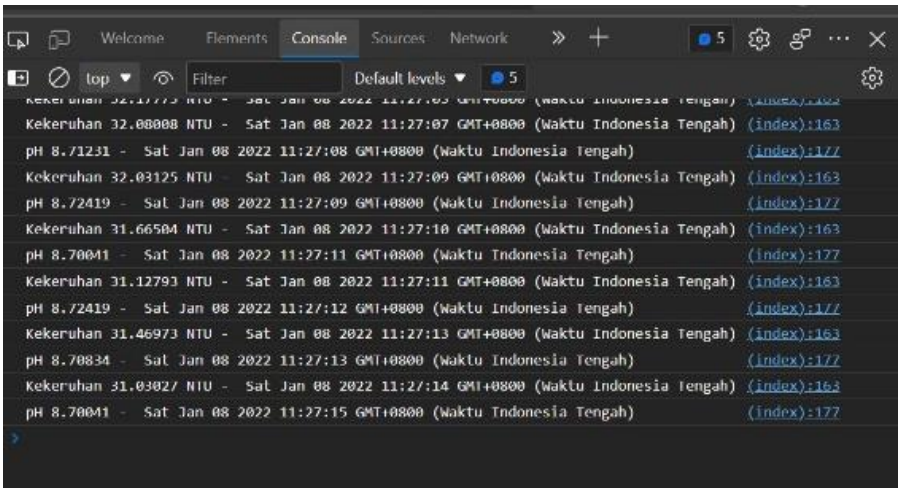
Gambar 4.18 Kecepatan internet (3)

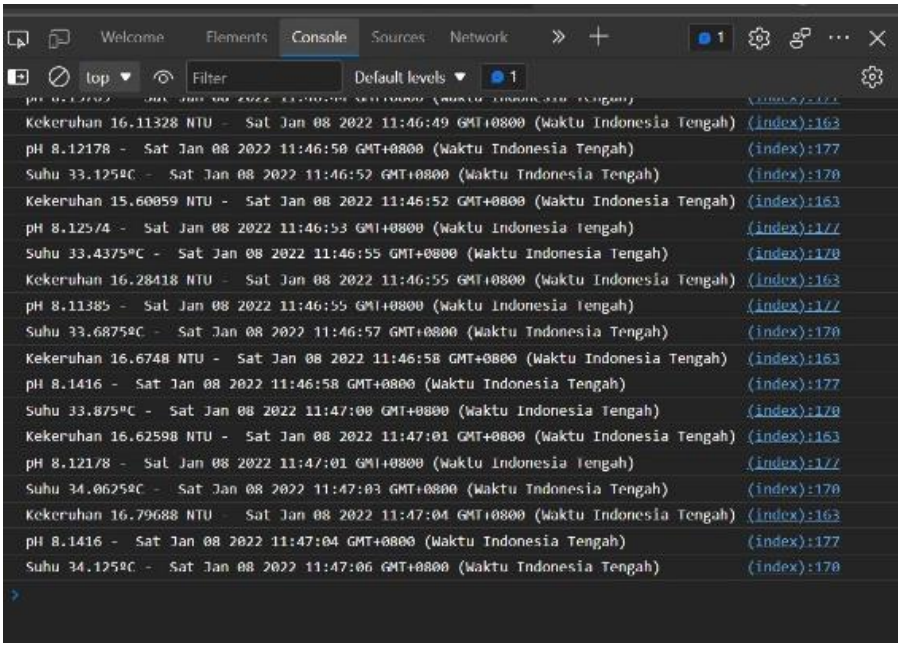
Tabel 4.13 Pengujian kecepatan pengiriman data sensor (3)

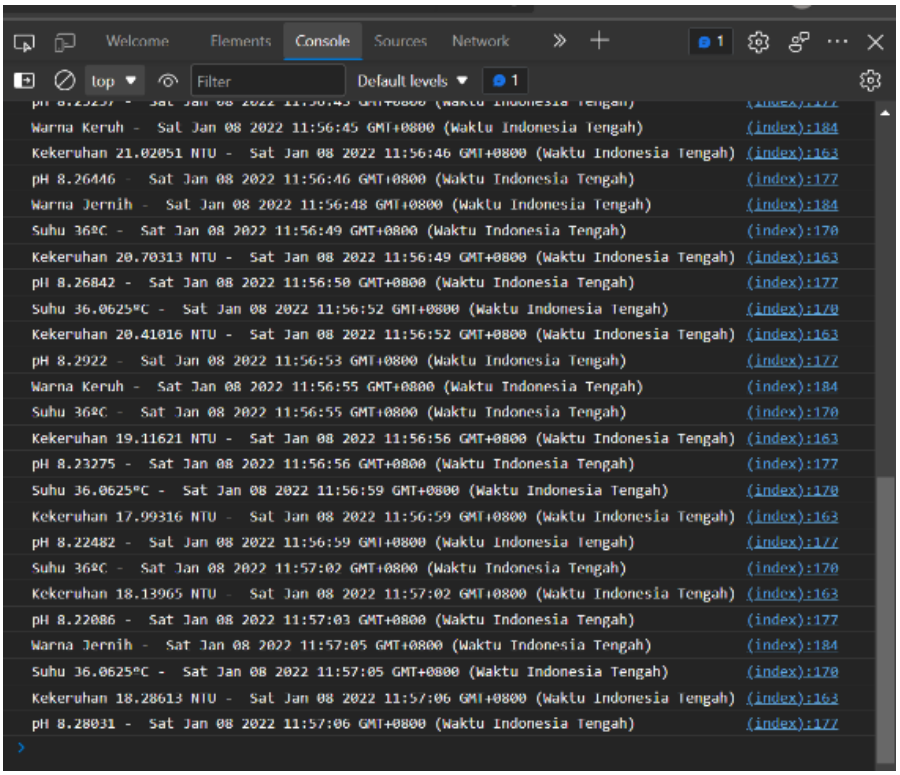
Sensor	Pengiriman data	Kecepatan
1 sensor	1	1 detik
	2	1 detik
	3	1 detik
	4	1 detik
	5	1 detik
Rata-rata		**1 detik**
2 sensor	1	1 detik
	2	2 detik
	3	1 detik
	4	1 detik
	5	2 detik
Rata-rata		**1.4 detik**
3 sensor	1	3 detik

Sensor	Pengiriman data	Kecepatan
3 sensor	2	2 detik
	3	3 detik
	4	3 detik
	5	3 detik
Rata-rata		**2.8 detik**
4 sensor	1	4 detik
	2	3 detik
	3	3 detik
	4	3 detik
	5	4 detik
Rata-rata		**3.4 detik**









Gambar 4.19 Pengiriman data sensor (3)

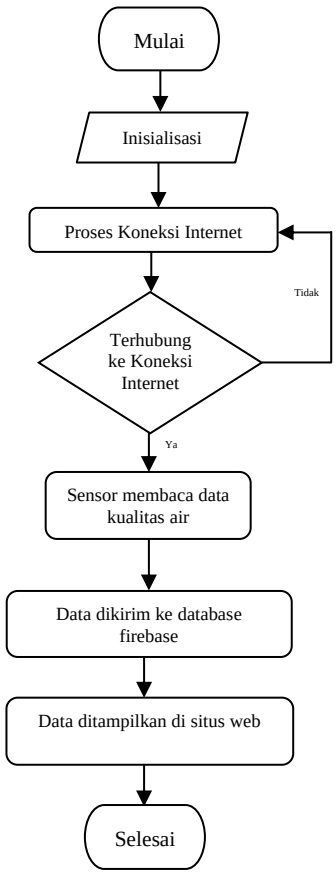
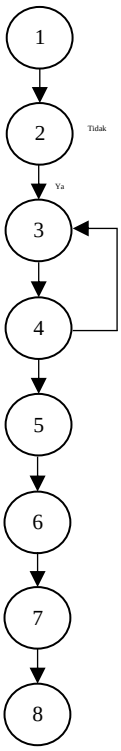
Pada tabel 4.13, pengujian tahap ketiga mendapatkan hasil menggunakan 1 (satu) sensor dengan nilai rata-rata 1 detik, 2 (dua)

sensor dengan nilai rata-rata 1.4 detik, menggunakan 3 (tiga) sensor dengan nilai kecepatan rata-rata 2.8 detik dan 4 (empat) sensor dengan nilai rata-rata 3.4 detik. Kecepatan internet pada tahap ketiga ini (*download* 41.82 Mbps dan *upload* 2.08 Mbps) serta nilai kecepatan rata-rata yang diperoleh menghasilkan bahwa jika semakin sedikit sensor yang digunakan, maka semakin sedikit juga waktu pengiriman data akan berlangsung. Semakin banyak sensor yang digunakan, maka waktu yang digunakan dalam pengiriman data juga akan bertambah.

Disimpulkan dari ketiga tahap pengujian kecepatan pengiriman data yang dilakukan, yang diperoleh dari waktu pengiriman data dari sensor ke database hingga ditampilkan ke situs web yang waktu pengirimannya dapat dilihat pada *console* seperti pada gambar 4.15, gambar 4.17 serta gambar 4.19 menghasilkan bahwa semakin sedikit sensor yang digunakan, maka semakin cepat juga pengiriman datanya. Sebaliknya, semakin banyak sensor yang digunakan, maka waktu pengiriman data akan bertambah. Dari ketiga kecepatan internet yang dipakai sudah cukup untuk melakukan akses dalam pengiriman datanya. Namun juga tentunya pengiriman data akan dipengaruhi oleh kecepatan internet yang sedang diakses dan jumlah data program yang digunakan.

3. Pengujian *white box*

Tabel 4.14 *Flowchart* dan *flowgraph* sistem

Flowchart	***Flowgraph***
	

Dari *Flowgraph* di atas dapat dilakukan proses perhitungan sebagai berikut:

a. Menghitung C*yclomatic Complexcity* V(G) dari *Egde* dan *Node*:

Dengan rumus : V (G) = E – N + 2

Dengan E (*edge*) = 8

N (*node*) = 8

P (Predikat *node*) = 1

Penyelesaian : V (G) = E – N + 2

= 8 – 8 + 2

= 2

Predikat *node* (P) = P + 1

= 1 + 1

= 2

b. Berdasarkan perhitungan C*yclomatic Complexcity* dari *Flowgraph* diatas memiliki *Region* = 2

c. *Independent Path* pada *Flowgraph* diatas adalah :

Path 1 = 1 – 2 – 4 – 3

Path 2 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8

d. Graph Matrix

	1	2	3	4	5	6	7	8	Koneksi
1		1							1-1= 0
2			1						1-1= 0
3				1					1-1= 0
4			1		1				2-1= 1
5						1			1-1= 0
6							1		1-1= 0
7								1	1-1= 0
8									

Cyclomatic Complexity= 1+1 =2

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan hasil pengujian yang telah dilakukan, dapat disimpulkan beberapa hal sebagai berikut:

1. Alat bekerja dengan baik, mulai dari pembacaan sensor hingga ditampilkannya nilai sensor pada situs web. Nilai atau data sensor yang ditampilkan juga dengan pembacaan keakuratan yang cukup baik.
2. Pada pengujian kecepatan pengiriman data, semakin sedikit sensor yang digunakan, maka semakin cepat waktu pengiriman data dilakukan. Sebaliknya, semakin banyak sensor yang digunakan, kecepatan pengiriman data juga akan bertambah. Pengujian dengan menggunakan kecepatan internet *download* 41.82 Mbps dan *upload* 2.08 Mbps, diperoleh menggunakan 1 (satu) sensor dengan nilai rata-rata 1 detik, 2 (dua) sensor dengan nilai rata-rata 1.4 detik, menggunakan 3 (tiga) sensor dengan nilai kecepatan rata-rata 2.8 detik dan 4 (empat) sensor dengan nilai rata-rata 3.4 detik.
3. Pengiriman data sensor memerlukan jaringan internet, jaringan internet bisa mempengaruhi kecepatan pengiriman data sensor ke *database* hingga data ditampilkan pada situs web. Juga jumlah data yang digunakan pun pada pemprograman alat bisa mempengaruhi kecepatan pengiriman data.
4. Untuk mengakses *monitoring* nilai sensor di situs web menggunakan halaman link *https://iotmultisensor.web.app*

B. Saran

Adapun saran untuk mengembangkan sistem yang penulis angkat ini untuk lebih baik ke depannya sebagai berikut:

1. Alat ini dapat dikembangkan lagi dengan menambahkan sensor yang mengukur kualitas air berupa zat kimia yang terkandung pada air selain dari pengukuran sensor yang terdapat pada implementasi IoT ini.
2. Khusus pada pendeteksian warna air, perlu penggunaan sensor warna yang lebih akurat yang dapat mendeteksi cahaya pada air jernih hingga keruh.
3. Dalam *user interface monitoring* nilai sensor, dapat dikembangkan dengan menerapkan sistem aplikasi android sebagai sistem *interfacenya*.

DAFTAR PUSTAKA

Afrizal N. Baharsyah. 2019. *Pengertian Internet of Things (IoT): Semua Hal yang Perlu Kamu Tahu* (*online*). (https://www.jagoanhosting.com/blog/pengertian-internet-of-things-iot/. Diakses 09 November 2020).

Agus Faudin. 2017. *Apa itu Module NodeMCU ESP8266* (*online*). (https://www.nyebarilmu.com/apa-itu-module-nodemcu-esp8266/. Diakses 09 November).

Ahmad Sabiq, Prabowo Nugroho Budisejati. 2017. *Sistem Pemantauan Kadar pH, Suhu dan Warna pada Air Sungai Melalui Web Berbasis Wireless Sensor Network. Jakarta Pusat.*

Anifatul Faricha, Dimas Adiputra, Isa Hafidz, Lora Khaula Amifia, Moch. Iskandar Riansyah. 2019. *Analisa Studi tentang Perancangan Alat Monitoring Kualitas Air PDAM Berbasis Internet of Things. Kota Surabaya.*

Arum Sutrisni Putri. 2020. *Pencemaran Air : Pengertian, Penyebab, Dampak, Pencegahan* (*online*). (https://www.kompas.com/skola/read/2020/01/15/170000969/pencemaran-air--pengertian-penyebab-dampak-pencegahan?page=all. Diakses 09 November 2020).

Sahlan Abdul Muzib. *Pengujian dan analisis. UNIKOM*

Scholastika Gerintya. 2018. *Bagaimana Mutu dan Akses Air Bersih di Indonesia?* (Online). (https://tirto.id/bagaimana-mutu-dan-akses-air-bersih-di-indonesia-cGrk. Diakses 06 November 2020).

Tedy Tri Saputro. 2017. *Mengenal NodeMCU: Pertemuan Pertama* (*online*). (https://embeddednesia.com/v1/tutorial-nodemcu-pertemuan-pertama/. Diakses 09 November 2020).

Trisiani Dewi Hendrawati, Adnan Rafi Al Tahtawi, Farhan Fadilah. 2018. *Sistem Monitoring Pencemaran Air Sungai Berbasis Teknologi Sensor Nirkabel dan Internet-of-Things. Sukabumi. Bandung.*

Yusriadi Tauhid. 2013. *Parameter Kualitas Air* (*online*). (https://blog.ub.ac.id/yusriadiblog/2013/11/04/parameter-kualitas-air/. Diakses 09 November 2020).