**SISTEM PEMANTAUAN DAN PEMELIHARAAN AKUARIUM**

**TEROTOMATISASI**



**TUGAS AKHIR**

*Disusun dalam rangka memenuhi salah satu persyaratan*

*Untuk menyelesaikan program Strata-1 Departemen Teknik Informatika*

*Fakultas Teknik Universitas Hasanuddin*

*Makassar*

**Disusun Oleh:**

**MUHAMMAD IRSYAD ASHARI**

**D421 16 305**

**DEPARTEMEN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS HASANUDDIN**

**2020**

# LEMBAR PENGESAHAN SKRIPSI

# ABSTRAK

Memelihara ikan hias merupakan hobi dan usaha yang sudah ditekuni oleh masyarakat indonesia sejak lama. Terlebih lagi mudahnya mendapatkan ikan hias yang dijual secara bebas di pasar tradisional maupun pasar modern yang sudah tersebar di berbagai lokasi di Indonesia. Dalam memelihara ikan hias terdapat faktor-faktor penting yang harus pemelihara ikan perhatikan dalam melakukan perawatan dan pemeliharaan ikan hias. Faktor tersebut adalah kondisi ideal kolam ikan yang harus memenuhi kondisi yang membuat ikan tersebut nyaman hidup di dalamnya. Karena faktor tersebutlah pemelihara ikan hias harus sering memantau dan melakukan perawatan terhadap kondisi kolam atau akuarium tempat ikan hias hidup.Salah satu solusi kondisi akuarium dapat dipantau dan dikelola dengan efisien adalah dengan memanfaatkan sensorberbasis *Internet of Things (IoT).*Penggunaan Raspberry Pi sebagai *Web server* dan mikrokontroler banyak dimanfaatkan oleh aplikasi berbasis *Internet of Things* karena penggunaan daya yang rendah dengan kemampuan komputasi yang sudah mencapai 700MHz hingga 1,4GHz.

Penelitian ini bertujuan untuk membangun sistem yang dapat memantau kondisi akuarium ikan hias serta melakukan penggantian otomatis terhadap air akuarium berbasis konsep *Internet of Things*. Penelitian ini memiliki fokus kerja memantau suhu, ketinggian, dan kekeruhan air , penggantian air otomatis dan pemantauan secara jarak jauh melalui media halaman web. Sensor yang digunakan adalah DS18B20, JSN-SR04T, ACS712 5A & SEN-0189. Keempat sensor tersebut akan disambungkan ke sebuah Raspberry Pi untuk diolah datanya. Data yang telah diolah tersebut akan di tunjukkan secara *realtime* di sebuah halaman web menggunakan *HTTP Request* sehingga dapat dipantau secara jarak jauh melalui halaman web.Berdasarkan pengujian sistem penggantian air secara otomatis yang telah dilakukan, diperoleh kesimpulan bahwa rata-rata akurasi ketinggian air setelah penggantian air sebesar 96,5% pada 2 kali percobaan.

**Kata kunci** : Sensor, *HTTP Request*, *Internet of Things,* Raspberry pi

# KATA PENGANTAR

Assalamu Alaikum Wr. Wb. Puji dan syukur penulis panjatkan atas kehadirat Allah SWT karena berkat Rahmat dan Karunia-Nya sehingga Tugas Akhir yang berjudul “**SISTEM PEMANTAUAN DAN PEMELIHARAAN AKUARIUM TEROTOMATISASI** ” ini dapat diselesaikan sebagai salah satu syarat dalam menyelesaikan jenjang Strata- 1 pada Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin.

Penyusunan penelitian ini disajikan hasil suatu penelitian yang menyangkut judul yang telah diangkat dan telah melalui proses pencarian dari berbagai sumber baik jurnal penelitian, buku maupun dari situs-situs di internet.

Penulis menyadari bahwa dalam penyusunan dan penulisan skripsi ini tidak lepas dari bantuan, bimbingan serta dukungan dari berbagai pihak, dari masa perkuliahan sampai dengan masa penyusunan tugas akhir, sangatlah sulit untuk menyelesaikan tugas akhir ini. Oleh karena itu, penulis dengan senang hati menyampaikan terima kasih kepada:

1. Kedua Orang tua penulis, Bapak Muhammad Ichsan Bayupah,SE,MM. dan Ibu Ratna Amir Samiallah serta keluarga besar penulis yang selalu memberikan dukungan, doa, dan semangat, juga selalu sabar dalam mendidik penulis sejak kecil;
2. Bapak Adnan, S.T., M.T., Ph.D., selaku pembimbing 1 dan Bapak Ir. Christoforus Yohannes, M.T., selaku pembimbing II yang selalu menyediakan waktu, tenaga, pikiran dan perhatian yang sangat luar biasa untuk mengarahkan penulis dalam penyusunan tugas akhir;
3. Bapak Dr. Ir. Amil Ahmad Ilham, ST., M.IT., selaku Ketua Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin atas bimbingannya selama masa perkuliahan penulis;
4. Kepada Fachrial Yunizar S.T., Rizky Eka Arlin , dan Tiwi Nur Safitri S.T., yang selalu membantu selama penelitian, dan diskusi proses penyusunan Tugas Akhir, serta menjadi mentor dalam penyusunan skripsi ini;
5. Kepada Nihlahtuzzahra, yang tidak henti-hentinya memberikan kasih sayang, semangat, dan bantuan selama penyusunan skripsi ini;
6. Kepada Nurwanti Aprilia yang selalu mendukung dalam hal teknis penulisan ilmiah.
7. Para stackholder dari PT. CARAKDE, yang tiada henti-hentinya memberikan dukungan, masukan dan semangat di masa-masa sulit;
8. Keluarga angkatan 2016 Departemen Teknik Informatika FT-UH atas semua bantuan dan semangat yang diberikan selama ini;
9. Teman-teman angkatan IGNITER FT-UH atas dukungan dan semangat yang diberikan selama ini;
10. Para Sahabat, teman-teman, adik-adik dan kakak-kakak anggota Labolatorium Internet of Things dan Parallel Computing Universitas Hasanuddin yang telah memberikan begitu banyak bantuan selama penelitian, pengambilan data dan diskusi progress penyusunan Tugas Akhir;
11. Segenap Staf Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin yang selalu dan tak kenal lelah membantu penulis;
12. Orang-orang berpengaruh lainnya yang tanpa sadar telah menjadi inspirasi penulis;

Akhir kata, penulis berharap semoga Allah SWT. berkenan membalas segala kebaikan dari semua pihak yang telah banyak membantu. Semoga Tugas Akhir ini dapat memberikan manfaat bagi pengembangan ilmu. Aamiin.

*Wassalam*

Makassar, 01 April 2020

Penulis

# DAFTAR ISI

**halaman**

**HALAMAN JUDUL ………………………………………………………………… i**

[LEMBAR PENGESAHAN SKRIPSI ii](#_Toc42280405)

[ABSTRAK iii](#_Toc42280406)

[KATA PENGANTAR iv](#_Toc42280407)

[DAFTAR ISI vii](#_Toc42280408)

[DAFTAR GAMBAR xi](#_Toc42280409)

[DAFTAR TABEL xiii](#_Toc42280410)

[BAB I PENDAHULUAN 1](#_Toc42280411)

[I.1 Latar Belakang 1](#_Toc42280413)

[I.2 Rumusan Masalah 3](#_Toc42280414)

[I.3 Tujuan Penelitian 3](#_Toc42280415)

[I.4 Batasan Masalah 4](#_Toc42280416)

[I.5 Manfaat Penelitian 5](#_Toc42280417)

[I.6 Sistematika Penulisan 5](#_Toc42280418)

[BAB II TINJAUAN PUSTAKA 7](#_Toc42280419)

[II.1 Ikan Hias 7](#_Toc42280421)

[II.2 Pemeliharaan Akuarium Ikan Hias 7](#_Toc42280422)

[II.2.1 Penerangan Pada Akuarium Ikan Hias 8](#_Toc42280423)

[II.3 Memantau Suhu Air Akuarium Secara Tradisional 9](#_Toc42280424)

[II.4 Metode Penggantian Air Ikan Hias Akuarium 10](#_Toc42280425)

[II.5 Sensor Suhu DS18B20 11](#_Toc42280426)

[II.6 Sensor Ultrasonik JSN-SR04T 12](#_Toc42280427)

[II.7 ADC Converter ADS1115 16 Bit 13](#_Toc42280428)

[II.8 Sensor Turbiditas SKU SEN0189 13](#_Toc42280429)

[II.9 Sensor Arus ACS712 5A 14](#_Toc42280430)

[II.10 Mini PC Raspberry Pi 15](#_Toc42280431)

[II.11.1 Sistem Operasi Raspberry Pi 16](#_Toc42280432)

[II.11.2 Raspberry Pi 4 Model B 18](#_Toc42280433)

[II.11.3 Spesifikasi Raspberry Pi 4 Model B 18](#_Toc42280434)

[II.12 Web Server 19](#_Toc42280435)

[II.12.1 Web Server NginX 21](#_Toc42280436)

[II.11.1.1 Keunggulan Web Server NginX 21](#_Toc42280437)

[II.12.1.2 Fasilitas Web Server NginX 22](#_Toc42280438)

[II.13 Port Forwarding 23](#_Toc42280439)

[II.13.1 NGroK 24](#_Toc42280440)

[BAB III METODOLOGI PENELITIAN 26](#_Toc42280441)

[III.1 Tempat dan Waktu Penelitian 26](#_Toc42280443)

[III.2 Alat dan Bahan Penelitian 26](#_Toc42280444)

[III.3 Kriteria Desain 27](#_Toc42280445)

[III.4 Prosedur Penelitian 28](#_Toc42280446)

[III.5 Tahap Persiapan 30](#_Toc42280447)

[III.6 Tahap Pembuatan Sistem 31](#_Toc42280448)

[III.6.1 Tahap Perancangan Sistem Hardware (Perangkat Keras). 31](#_Toc42280449)

[III.6.1.1 Raspberry Pi 4 Model B 33](#_Toc42280450)

[III.6.1.2 Sensor Suhu *Waterproof* DS18B20 33](#_Toc42280451)

[III.6.1.3 ADC Konverter ADS1115 16 Bit 34](#_Toc42280452)

[III.6.1.3 Sensor Ultrasonik Waterproof JSN-SR04T 35](#_Toc42280453)

[III.6.1.4 Sensor Turbiditas DFRobot SEN-0189 36](#_Toc42280454)

[III.6.1.5 Sensor Arus ACS712 5A 37](#_Toc42280455)

[III.6.1.6 Relay 4 *channel* 5V 38](#_Toc42280456)

[III.6.2 Tahap Perakitan / Perwujudan Alat 39](#_Toc42280457)

[III.6.3 Tahap Pembuatan Software (Perangkat Lunak) 40](#_Toc42280458)

[III.6.3.1 Sistem Penggantian air otomatis 42](#_Toc42280459)

[III.6.3.2 Menghitung Penggunaan Daya 44](#_Toc42280460)

[III.6.3.3 Visualisasi Data ke Halaman Web 45](#_Toc42280461)

[BAB IV HASIL DAN PEMBAHASAN 46](#_Toc42280462)

[IV.1 Hasil Rancang Bangun Alat 46](#_Toc42280463)

[IV.2 Pengujian Sistem 49](#_Toc42280464)

[IV.2.1 Pengujian Sistem Pembacaan Suhu 49](#_Toc42280465)

[IV.2.2 Pengujian Sistem Pembacaan Ketinggian Air 51](#_Toc42280466)

[IV.2.3 Pengujian Sistem Pembacaan Kekeruhan Air 52](#_Toc42280467)

[IV.2.4 Pengujian Sistem Penggantian Air Otomatis 52](#_Toc42280468)

[IV.2.4.1 Pengujian Sistem Penggantian Air Otomatis Metode Waktu 54](#_Toc42280469)

[IV.2.4.2 Pengujian Sistem Penggantian Air Otomatis Metode Kekeruhan Air 57](#_Toc42280470)

[IV.2.5 Pengujian Sistem Pemantauan Data Pada Web Browser 59](#_Toc42280471)

[IV.3 Hasil Pengamatan dan Data 62](#_Toc42280472)

[IV.3.1 Halaman Web 62](#_Toc42280473)

[IV.3.2 Penggantian Air Otomatis 64](#_Toc42280474)

[BAB V PENUTUP 66](#_Toc42280475)

[V.1 Kesimpulan 66](#_Toc42280477)

[V.2 Saran 67](#_Toc42280478)

[DAFTAR PUSTAKA 69](#_Toc42280479)

# 

# DAFTAR GAMBAR

**Gambar 2.1** Penampakan Termometer Suhu Akuarium …………………….... 10

**Gambar 2.2** Konfigurasi Pin DS18B20……………………………………….. 11

**Gambar** **2.3** Komponen Raspberry Pi 3 Model B …………………………….. 18

**Gambar** **2.4** UI Konsol NGroK ..........................…..………………………….. 25

**Gambar 3.1** Gambaran umum rangkaian sistem …………………………….... 27

**Gambar 3.2** Diagram tahapan penelitian ……………………………………… 29

**Gambar 3.3** Rangkaian sistem ………………………………………………… 31

**Gambar 3.4** Id dari kedua buah sensor DS18B20 …………………………….. 34

**Gambar 3.5** Nilai Balik Sensor Arus Ketika Arus Listrik Terdeteksi ...…….... 37

**Gambar 3.6** Nilai Balik Sensor Arus Ketika Arus Listrik Tidak Terdeteksi .... 37

**Gambar 3.7** Blok diagram Cara Kerja Alat ........................................................ 40

**Gambar 3.8** *Flowchart* sistem pemantauan akuarium ........................................ 41

**Gambar 3.9**Flowchart Sistem penggantian air otomatis .................................... 43

**Gambar 4.1** Box prototype tempat penyimpanan komponen elektronika ........... 46

**Gambar4.2** Penempatan sensor ultrasonik untuk mengukur ketinggian air …... 47

**Gambar 4.3**Penempatan sensor suhu di air akuarium cadangan ……………... 48

**Gambar 4.4**Penempatan sensor turbiditas di filter air akuarium ....................... 48

**Gambar 4.5**Tampilan Data hasil pembacaan suhu air akuarium pada terminal 49

**Gambar 4.6**Hasil Pembacaan keitnggian air akuarium .....................................50

**Gambar 4.7** Jadwal disetel untuk mengganti air tiap hari senin ......................... 54

**Gambar 4.8**Kondisi akuarium penuh saat air belum diganti ............................. 54

**Gambar 4.9** Kondisi akuarium saat air sedang diganti ....................................... 55

**Gambar 4.10**Akuarium saat proses penggantian air telah selesai dieksekusi .................. 55

**Gambar 4.11**Nilai tegangan air keruh ............................................................... 57

**Gambar 4.12** Air akuarium keruh ...................................................................... 57

**Gambar 4.13** Kondisi akuarium saat air sedang diganti .................................... 58

**Gambar 4.14**Akuarium saat proses penggantian air telah selesai dieksekusi ................. 58

**Gambar 4.15** Pemantauan Data pada Website ................................................... 61

**Gambar 4.16** Halaman login .............................................................................. 63

**Gambar 4.17**User request ke server setiap 15 detik ..........................................63

# DAFTAR TABEL

**Tabel 2.1** Spesifikasi Teknis JSN-SR04T ........................................................... 12

**Tabel 2.2** Konfigurasi pin sensor ACS712 .......................................................... 15

**Tabel 3.1** Daftar komponen yang digunakan dalam sistem ................................. 32

**Tabel 3.2** Wiring antara DS18B20 dan Respberry Pi .......................................... 34

**Tabel 3.3** Wiring antara ADS1115 dan Sensor Analog ....................................... 35

**Tabel 3.4** Wiring antara ADS1115 dan Raspberry pi .......................................... 35

**Tabel 3.5** Wiring antara JSN-SR04T dan Raspberry pi ....................................... 36

**Tabel 3.6** sambungan kanal relay dengan GPIO pins .......................................... 39

**Tabel 4.1** Data Akurasi pembacaan sensor ketinggian air ................................... 64

# 

# BAB I

# PENDAHULUAN

## I.1 Latar Belakang

Memelihara ikan hias adalah salah satu hobi yang banyak digandrungi oleh orang Indonesia. Salah satu wadah yang biasanya dipilih oleh pemelihara ikan hias adalah akuarium karena permukaan dindingnya yang transparan sehingga penampakan ikan hias dapat dinikmati. Untuk memelihara ikan hias para pemelihara akan senantiasa merawat akuarium yang menjadi habitat dari ikan hias kesayangannya agar kesehatan ikannya terjaga. Perawatan yang dilakukan pada akuarium biasanya seperti mengganti kapas filter, memberi penerangan yang cukup pada akuarium, serta mengganti air akuarium secara rutin dengan jadwal yang konsisten.

Dalam proses perawatan akuarium, tantangan yang paling sering dihadapi oleh pemelihara ikan hias ialah memastikan suhu air tetap stabil, menjaga agar penerangan akuarium cukup (menyalakan lampu akuarium tidak lebih dari 12 jam sehari), dan yang paling memberatkan ialah melakukan proses penggantian air yang memakan waktu yang tidak sedikit.

Penulis sendiri telah melakukan uji coba langsung terhadap akuarium berukuran 90x50x50 cm**.** Ketinggian air adalah 40 cm, jadi volume air adalah 90 x 40 x 50 / 1000 = 180 liter.Untuk menyedot 30% volume air (54 liter) dengan pompa berkapasitas 5 liter / menit, dibutuhkan waktu 10 menit 48 detik hanya untuk menyedot air kotor dan membutuhkan 10 menit 48 detik lagi untuk mengisi air kembali ke volume semula sehingga total waktu hanya untuk menyedot dan mengisi air akuarium adalah 21 menit 36 detik. Jumlah akuarium yang banyak tentu saja akan membuat pemelihara ikan harus menghabiskan lebih banyak waktu lagi hanya untuk mengganti air.

Proses penggantian air yang bersifat berulang ini mendorong peneliti agar merancang sebuah sistem yang dapat mengotomatisasi perawatan akuarium di atas agar keindahan ikan tetap dapat dinikmati dengan usaha perawatan yang telah diminimalisir. Sistem otomatisasi yang akan dibuat juga akan dibuatkan sebuah fitur pemantauan jarak jauh agar pemelihara ikan tetap dapat memantau kondisi akuariumnya dari jarak jauh ketika sedang tidak berada di lokasi dikarenakan sedang mudik atau sedang meninggalkan rumah dalam jangka waktu yang lama.

Hal ini sangat mungkin untuk diwujudkan melihat perkembangan teknologi saat ini. Mikrokontroller telah berkembang menjadi semakin cepat, penggunaan daya yang rendah, koneksi nirkabel, serta yang paling penting ialah mikrokontroller telah menjadi semakin kecil yang memungkinkannya untuk disematkan ke benda mati termasuk komponen elektronik di akuarium agar dapat di pantau dari jarak jauh melalui internet. Sistem yang akan dibuat oleh peneliti akan mengintegrasikan semua sensor-sensor dan aktuator untuk mengontrol komponen-komponen elektronika yang mendukung kehidupan di dalam akuarium seperti pompa utama dan *heater .* Konsep ini telah menjadi tren dengan nama *Internet of Things* (IoT). IoT adalah jaringan benda fisik khusus yang mengandung teknologi tersemat untuk berkomunikasi dan merasakan atau berinteraksi dengan keadaan internal mereka atau lingkungan eksternal. Penelitian ini bertujuan untuk membuat sistem pemantauan dan otomatisasi penggantian air akuarium dengan pemanfaatan teknologi berdaya rendah. Hasil rancang sistemnya diharapkan mampu membantu para pemelihara ikan hias untuk menjadikan perawatan ikan hias di akuarium menjadi lebih mudah.

## I.2 Rumusan Masalah

Rumusan masalah yang akan diuraikan dalam tugas akhir ini antara lain:

a. Bagaimana merancang sistem penggantian air secara otomatis dengan mempertahankan volume air di dalam akuarium ?

b. Bagaimana data-data sensor analog dan digital yang dipasang di akuarium diolah di web server ?

c. Bagaimana merancang sistem pemantauan jarak jauh dengan media halaman web ?

d. Bagaimana menghitung sisa waktu penggunaan listrik dengan menghitung penggunaan daya yang digunakan oleh sistem dalam mode baterai ketika listrik mati ?

## I.3 Tujuan Penelitian

Tujuan akhir dari penelitian ini antara lain:

1. Untuk membuat sebuah sistem tersemat pada akuarium yang dapat melakukan penggantian air secara otomatis dan menjaga ketinggian air di dalam akuarium tetap sama.
2. Untuk membuat sebuah sistem pemantauan berbasis web pada sebuah akuarium yang dapat diakses dari lokasi mana saja menggunakan perangkat komputer yang terhubung ke internet dan memiliki web *browser*.
3. Untuk membuat sebuah sistem tersemat pada akuarium yang dapat tetap dipantau walaupun berjalan dengan waktu terbatas menggunakan baterai *UPS* ketika terjadi pemadaman listrik.

## I.4 Batasan Masalah

Yang menjadi batasan masalah dalam tugas akhir ini adalah:

1. Objek penelitian berupa akuarium berukuran 90x50x50 cm.
2. Sumber energi utama untuk sistem ini berupa baterai *UPS* yang selalu tersambung ke aliran listrik*.*
3. Pengambilan data dilakukan di sebuah akuarium berisi ikan

*Amphilophus trimaculatus*(Louhan), *Amphilophus Citrinellus* (Red parrot ) dan *Cichlasoma nigrofasciatum* (Chichild Tattoo).

1. Pengambilan data dilakukan menggunakan sensor digital dan

analog.

1. Target akuarium untuk mengimplementasikan sistem tersemat ini adalah akuarium yang hanya berisi ikan hias dan tidak berisi tanaman, pasir, batu atau hiasan akuarium apapun.
2. Otomatisasi yang di buat adalah proses penggantian air dengan mempertahankan volume air ideal bagi ukuran akuarium.

## I.5 Manfaat Penelitian

Dengan dilakukannya penelitian ini, diharapkan manfaat yang didapatkan antara lain:

* 1. Bagi pemelihara ikan hias, penelitian ini dapat digunakan untuk memberi keamanan terhadap aset mereka yang berupa ikan hias agar harapan hidupnya naik apabila terjadi hal-hal yang mengharuskan pemilik akuarium meninggalkan rumah atau bepergian ke tempat yang jauh dari akuariumnya.
  2. Bagi peneliti, penelitian ini dapat digunakan untuk menambah pengetahuan dan kemampuan di bidang *Internet of Things* (IoT) dalam bagaimana mengotomatisasi akuarium agar dapat beroperasi dengan campur tangan manusia yang minim.
  3. Bagi institusi pendidikan, penelitian ini dapat digunakan sebagai referensi ilmiah untuk penelitian-penelitian selanjutnya.

## I.6 Sistematika Penulisan

Untuk memberikan gambaran singkat mengenai isi tulisan secara keseluruhan, maka akan diuraikan beberapa tahapan dari penulisan secara sistematis, yaitu :

**BAB I PENDAHULUAN**

Bab ini menguraikan secara umum mengenai hal yang menyangkut latar belakang, perumusan masalah dan batasan masalah, tujuan, manfaat, dan sistematika penulisan.

**BAB II TINJAUAN PUSTAKA**

Bab ini berisi teori-teori tentang hal-hal yang berhubungan dengan budidaya ikan hias, pembacaan sensor pada Mini PC raspberry pi, *Internet of Things*, pembuatan *web server* pada raspberry pi dan metode yang digunakan.

**BAB III METODOLOGI PENELITIAN**

Bab ini membahas tentang jenis penelitian, alat dan bahan penelitian, metode perancangan sistem, metode pengujian dan hasil penelitian.

**BAB IV HASIL DAN PEMBAHASAN**

Bab ini berisi tentang hasil perancangan sistem, pengujian sistem dan pengolahan data hasil penelitian.

**BAB V PENUTUP**

Bab ini berisi tentang kesimpulan yang didapatkan berdasarkan hasil penelitian yang telah dilakukan serta saran-saran untuk pengembangan lebih lanjut.

# BAB II

# TINJAUAN PUSTAKA

Seperti pada penjelasan sistematika penulisan pembuatan tugas akhir ini, dalam bab ini akan dijelaskan beberapa landasan-landasan teori sebagai hasil dari studi literatur yang berhubungan dalam perancangan dan pembuatan alat. Pada bab ini penulis akan membahas dasar-dasar dari beberapa bagian penting yang terdapat pada sistem otomatisasi dan pemantauan akuarium berbasis *IoT*.

## II.1 Ikan Hias

Menurut Badan Pengembangan Ekspor Nasional (BPEN) diacu dalam Kusniati N (2007), ikan hias adalah ikan yang umumnya mempunyai bentuk, warna, dan karakter yang khas, sehingga mampu memberikan suasana yang mendukung tata ruang serta mampu memberikan suasana “tentram dan nyaman”.

## II.2 Pemeliharaan Akuarium Ikan Hias

Hobi pemeliharaan ikan hias telah berkembang jauh selama dekade terakhir, sebagian berkat kemajuan peralatan akuarium dan banyaknya informasi yang tersedia. Ada banyak buku ikan dan akuarium yang luar biasa tersedia serta banyak informasi di internet, forum dan grup diskusi.

Perawatan akuarium atau pemeliharaan tangki ikan adalah sesuatu yang mutlak harus dilakukan secara teratur untuk mendapatkan hasil maksimal dari hobi ini. Anda akhirnya membuat tangki ikan Anda berjalan tetapi akan ada saatnya Anda memperhatikan bahwa itu mulai terlihat sedikit kotor. Atau, Anda perhatikan bahwa Anda perlu mengisi air yang menguap. Ini adalah waktu untuk perawatan atau pemeliharaan akuarium rutin. Perawatan akuarium atau pemeliharaan tangki ikan kadang-kadang bisa mengganggu bagi penghobi tetapi tidak harus selalu seperti itu. Kembangkan jadwal untuk melakukan tugas-tugas pemeliharaan akuarium ini. Penggantian air itu harus meningkatkan kesehatan ikan Anda dan membuat tangki Anda terlihat lebih bagus.

Penting untuk dicatat bahwa Anda tidak perlu mengganti air tangki sepenuhnya setiap kali Anda harus "membersihkan tangki" atau melakukan pemeliharaan akuarium. Sebagian besar Anda hanya perlu melakukan perubahan air parsial (sekitar 30 persen volume air). Jika Anda memiliki masalah dengan air keruh, Anda mungkin memberi makan terlalu banyak, tangki Anda mungkin terlalu banyak ikan, Anda tidak cukup sering melakukan penggantian air, Anda memberi makan jenis makanan yang salah, dll. Atau, itu bisa merupakan kombinasi dari yang di atas (FishLore : Freshwater Aquarium Book, 2013).

### II.2.1 Penerangan Pada Akuarium Ikan Hias

Pencahayaan akuarium yang tepat sangat penting untuk menjaga kesehatan ikan di akuarium. Pencahayaan yang baik juga bisa membuat akuarium jadi lebih hidup. Akuarium ikan hias yang indah bisa menjadi daya tarik tersendiri di rumah dan juga bisa menyatukan keluarga. Oleh karena itu, pemelihara ikan hias harus memikirkan pencahayaan yang baik untuk akuarium.

Pencahayaan memainkan peranan penting dalam menciptakan pemandangan akuatik yang indah, tetapi hal yang tak kalah pentingnya untuk disadari adalah akuarium yang berbeda memiliki kebutuhan pencahayaan yang berbeda, ikan yang berbeda juga memiliki kebutuhan pencahayaan yang berbeda.

Semakin kecil akuarium yang dimiliki, maka semakin rentan terhadap kenaikan suhu yang tajam akibat cahaya panas. Meninggalkan lampu akuarium menyala dalam waktu lama atau bahkan dalam semalam terbukti bisa mematikan bagi para makhluk penghuni akuarium, oleh karena itu perlu dimatikan.

Memantau suhu air akuarium secara berkala diperlukan untuk mengetahui berapa lama lampu akuarium dapat dinyalakan dengan aman. Suhu rumah, lokasi akuarium di rumah (misalnya di dekat jendela), ukuran akuarium, serta jenis ikan dan tanaman semuanya menjadi faktor penentu dalam mengambil keputusan. Jangan pernah biarkan lampu akuarium menyala selama 24 jam penuh karena di alam liar ikan juga membutuhkan siklus antara pagi dan malam. Penerangan yang disarankan hanya dilakukan selama maksimal 12 jam saja setiap hari. (Larry McGee, 2018)

## II.3 Memantau Suhu Air Akuarium Secara Tradisional

Selama ini para pemelihara ikan hias hanya menempelkan sebuah termometer yang menempel di kaca untuk memantau suhu air di dalam akuarium. Cara ini adalah cara tradisional yang memiliki kekurangan salah satunya yaitu untuk mengetahui suhu air, pemilik ikan diharuskan untuk melihat langsung suhu air dari depan akuarium .

Saat memelihara ikan hias, selalu perhatikan suhu dan kualitas air, dan perilaku yang tidak biasa. Dengan begitu Anda dapat membiakkan ikan lagi dengan mudah dengan menciptakan kondisi yang menyerupai habitat aslinya (FishLore : Freshwater Aquarium Book, 2013).



**Gambar 2.1** Penampakan Termometer Suhu Akuarium

(pro-shrimp.co.uk)

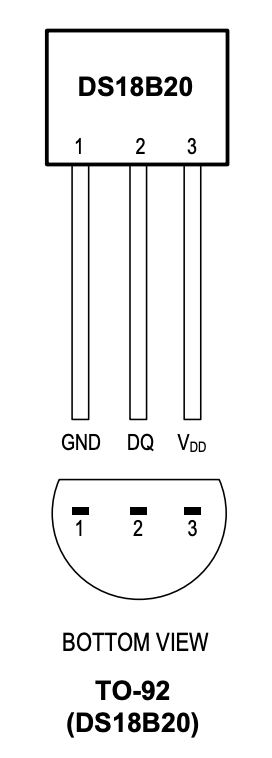
## II.4 Metode Penggantian Air Ikan Hias Akuarium

Salah satu cara untuk menghilangkan limbah ini di akuarium adalah melalui penggantian air secara parsial. Jumlah air yang perlu diubah tergantung pada berapa banyak dan seberapa besar ikan di akuarium, tetapi jumlah umum adalah 25% hingga 30% dari volume air per minggu. Menambahkan air yang telah menguap tidak termasuk mengganti air dikarenakan limbah terlarut tidak menguap. Saat menambahkan air baru, suhu air baru yang akan dimasukkan haruslah sedekat mungkin dengan suhu air di akuarium. Jika Anda tidak bisa mendapatkan air mendekati suhu akuarium, ganti sedikit air saja sehingga tidak akan mengagetkan ikan (Larry McGee, 2018).

## II.5 Sensor Suhu DS18B20

DS18B20 adalah termometer digital yang menyediakan penghitungan temperatur celsius dalam bentuk 9-12 Bit. DS18B20 berkomunikasi lebih dari 1-Wire bus yang menurut definisi hanya membutuhkan satu jalur data (dan *ground*) untuk komunikasi dengan pusat mikroprosesor.

Setiap DS18B20 memiliki kode serial 64-bit yang unik memungkinkan beberapa DS18B20 berfungsi pada bus 1-Wire yang sama. Jadi, mudah untuk menggunakan satu mikroprosesor untuk mengontrol banyak DS18B20 yang didistribusikan di area yang luas.

**Gambar 2.2** Konfigurasi Pin DS18B20

(<https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>)

## II.6 Sensor Ultrasonik JSN-SR04T

JSN-SR0T4-2.0 merupakan modul pengukuran jarak ultrasonik dapat menyediakan 20 cm – 600 cm jarak non-kontak. Fungsi sensor dengan tingkat akurasi hingga 2mm. Modul termasuk *transceiver* ultrasonik terintegrasi sensor dan rangkaian kontrol. Mode sekali pakai dan modul divisi JSN-SR04T-2.0. Produk ini mengadopsi desain *probe* ultrasonik terintegrasi tingkat industri, tipe tahan air, kinerja stabil dan jarak pengukuran akurat.

**Tabel 2.1** Spesifikasi Teknis JSN-SR04T

|  |  |
| --- | --- |
| Spesifikasi | Output Tegangan / Output Serial |
| Tegangan Operasi | DC 3.0-5.5V |
| Arus Kerja | < 8mA |
| Probe frequency | 40KHz |
| Jarak Maksimal | 600cm |
| Jarak Minimum | 20cm |
| Akurasi jarak | +- 1cm |
| Resolusi | 1mm |
| Sudut Pengukuran | 75 derajat |
| Sinyal Trigger | 1,10uS diatas tegangan TTL 2, port serial untuk mengirim instruksi 0X55 |
| Output sinyal echo | Tegangan output dengan level sinyal / TTL |
| Konfigurasi  pin | 3-5.5V (power positive)  Trig (RX) RX Echo (output)  TX GND (power supply negative |
| Ukuran produk | L42 \* W29 \* H12 mm |
| Temperatur kerja | -20 ° C to + 70 ° C |
| Warna Produk | PCB board berwarna biru |

## II.7 ADC Converter ADS1115 16 Bit

ADS1115 adalah modul ADC 16-bit yang presisi dengan empat *input* multipleks. Anda dapat menggunakan masing-masing *input* sendiri, atau berpasangan untuk pengukuran diferensial. Ini memiliki referensi kalibrasi internal untuk akurasi tinggi. Modul ini dapat berjalan dengan sinyal daya dan logika antara 2v hingga 5v, sehingga kompatibel dengan semua modul 3.3v dan 5v yang umum. Karena banyak dari 4 papan ini dapat dikontrol dari bus *2-wire* *I2C* yang sama, memberi Anda hingga 16 saluran diferensial tunggal atau 8. Penguat penguatan yang dapat diprogram menyediakan penguatan hingga x16 untuk sinyal kecil.

## II.8 Sensor Turbiditas SKU SEN0189

Sensor turbiditas mendeteksi kualitas air dengan mengukur tingkat kekeruhan. Menggunakan cahaya untuk mendeteksi partikel terendap dalam air dengan mengukur transmisi cahaya dan tingkat hamburan, yang berubah dengan jumlah total padatan tersuspensi (TSS) dalam air. Dengan meningkatnya TTS, cairan tingkat kekeruhan meningkat. Sensor kekeruhan digunakan untuk mengukur kualitas air di sungai dan aliran, air limbah dan limbah pengukuran, instrumentasi kontrol untuk kolam pengendapan, penelitian transportasi sedimen dan pengukuran laboratorium. Sensor ini menyediakan mode keluaran sinyal analog dan digital. Ambang disesuaikan saat mode sinyal digital.

Spesifikasi :

* Tegangan Pengoperasian: 5V DC
* Operasi Saat Ini: 40mA (MAX)
* Waktu Respon: <500ms
* Resistansi Isolasi: 100M (Min)
* Metode *Output*:
* *Output* analog: 0-4.5V
* *Output* Digital: Sinyal level Tinggi / Rendah (Dapat di sesuaikan dengan potensiometer)
* Berat: 30g
* Dimensi Adaptor: 38mm \* 28mm \* 10mm / 1.5 inci \* 1.1 inci \* 0.4 inci

## II.9 Sensor Arus ACS712 5A

Modul ACS712 menggunakan IC ACS712. Modul ini mendapatkan namanya dari IC (ACS712) yang digunakan dalam modul, jadi produk akhirnya sebenarnya menggunakan IC secara langsung, bukan modul.

Modul ACS712 ini dapat mengukur arus AC atau DC mulai dari + 5A hingga -5A, + 20A hingga -20A dan + 30A hingga -30A. Anda harus memilih rentang yang tepat untuk proyek Anda karena Anda harus menukar akurasi untuk modul rentang yang lebih tinggi. Modul ini menghasilkan tegangan Analog (0-5V) berdasarkan arus yang mengalir melalui kabel; oleh karena itu sangat mudah untuk menghubungkan modul ini dengan mikrokontroler. Jadi jika Anda mencari modul untuk mengukur arus menggunakan mikrokontroler untuk proyek Anda maka modul ini mungkin menjadi pilihan yang tepat untuk Anda. Adapun konfigurasi pin dari sensor ini dapat dilihat di tabel 2.2.

**Tabel 2.2** Konfigurasi pin sensor ACS712

|  |  |  |
| --- | --- | --- |
| **Nomor Pin** | **Nama Pin** | **Deskripsi** |
| 1 | VCC | Sambungan 5V |
| 2 | Data | *Output* tegangan analog yang mewakili besar arus |
| 3 | GND | Sambungan Ground |
| T1 | Kabel Masuk | Kabel yang dilalui arus harus diukur terhubung di sini |
| T2 | Kabel Keluar |

## II.10 Mini PC Raspberry Pi

Raspberry Pi merupakan sebuah komputer berukuran mini sebesar kartu kredit dengan harga yang relatif murah. Raspberry ini memiliki dua model yaitu Model A dan Model B. Perbedaan di antara keduanya hanya terletak pada keberadaan *Ethernet* yang absen pada model A dan jumlah port USB yang menjadi dua kalinya pada model B. Walaupun kecil dan murah, tetapi raspberry pi tidak bisa disebut murahan. Pasalnya, banyak karya dan kegunaan yang bias dihasilkan dari raspberry pi, mulai dari fungsi utamanya sebagai komputer yang memungkinkan kita untuk *browsing*, membuat laporan tugas, membuat *slide* presentasi, bermain game, atau sekedar mendengarkan musik dan menonton video. Raspberry pi pun sangat membantu dalam pembuatan karya-karya inovatif, seperti penggunaan raspberry pi dalam robot mata-mata yang dilengkapi kamera. Bahkan ada yang menjadikan raspberry pi sebagai pengendali mobil otomatis. Ada pula yang membuat raspberry pi ini menjadi *supercomputer*.

Raspberry terdiri dari beberapa model yaitu :

1. Raspberry Pi Model A
2. Raspberry Pi Model A+
3. Raspberry Pi Model B
4. Raspberry Pi Model B+
5. Raspberry Pi 2
6. Raspberry Pi 3 Model B+
7. Raspberry Pi 4 Model B+

## II.11.1 Sistem Operasi Raspberry Pi

Ini adalah daftar sistem operasi yang berjalan pada Raspberry Pi.

1. Full OS :

a. AROS

b. Haiku

c. LINUX :

- Android : Android 4.0 (Ice Cream Sandwich)

- Arch Linux ARM

- R\_Pi Bodhi Linux

- Debian Squeeze

- Firefox OS

- Gentoo Linux

- Google Chrome OS : Chromium OS

- PiBang Linux

- Raspberry Pi Fedora

- Raspbian

- Slackware ARM

- QtonPi

- WebOS : Open webOS

2. Multi-purpose light distributions:

* + 1. Moebius, ARMHF distribusi berdasarkan Debian.
    2. Squeezed Arm Puppy, versi Puppy Linux untuk ARM v6

3.Single-purpose light distributions:

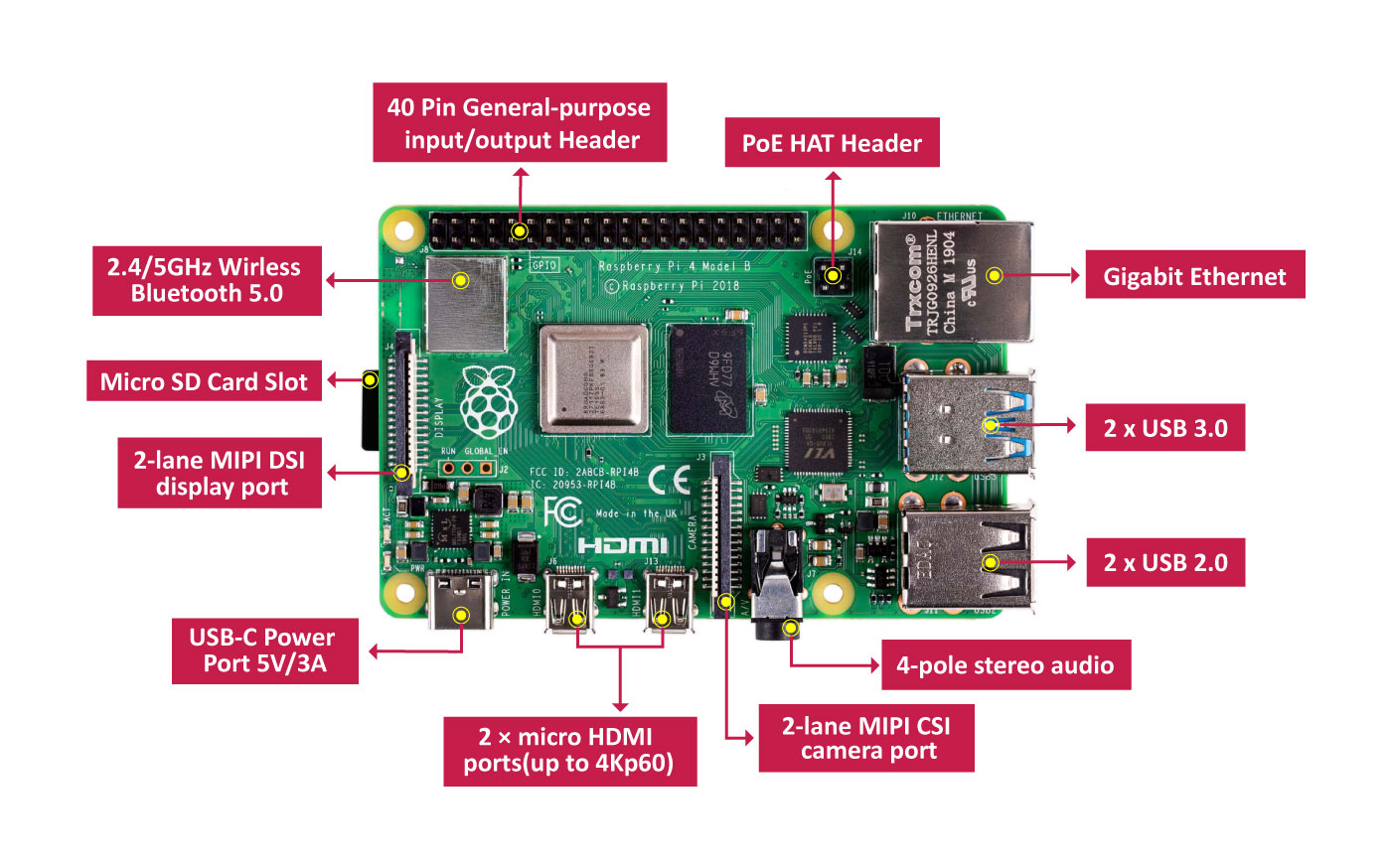
a.IPfire  
b.OpenELEC

c.Raspbmc

d.XBMaC  
e. XBian

## II.11.2 Raspberry Pi 4 Model B

Raspberry Pi 4 Model B adalah generasi keempat dari raspberry Pi. raspberry pi model ini dapat digunakan untuk banyak aplikasi dan menggantikan Model Raspberry Pi B+ dan Raspberry Pi 3 Model B. Raspberry Pi 4 Model B membawa prosesor yang lebih kuat dan 10 kali lebih cepat dibandingkan dengan generasi pertama dari Raspberry Pi. Selain itu Raspberry Pi model ini juga menambahkan konektivitas Wireless LAN dan Bluetooth (Damayanti & Fradita, 2017). Komponen pada raspberry pi 4 dapat dilihat pada gambar 2.3.



**Gambar** **2.3** Komponen Raspberry Pi 4 Model B

(www.seeedstudio.com)

## II.11.3 Spesifikasi Raspberry Pi 4 Model B

Adapun spesifikasi dari Raspberry Pi 4 Model B sebagai berikut :

* 1. Chipset Broadcom BCM2711
  2. Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
  3. 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
  4. 4 GB RAM
  5. 64 Bit CPU 64
  6. 4 x USB Ports
  7. 4 Pole Stereo output dan Composite video port
  8. 2 × micro-HDMI ports (up to 4kp60 supported)
  9. 10/100 BaseT Ethernet Socket
  10. Port kamera CSI untuk menghubungkan kamera Raspberry Pi
  11. Port tampilan DSI untuk menghubungkan tampilan layar sentuh Raspberry Pi
  12. Port Micro SD untuk memuat system operasi dan menyimpan data
  13. Sumber daya USB Tipe C

## II.12 Web Server

Web adalah tampilan pada browser dengan alamat domain khusus untuk sistem penelitian ini. Web dapat dibangun dengan menggunakan bahasa HTML dan PHP dengan gaya tampilan menggunakan bahasa CSS. Web tersebut disimpan pada satu komputer yang disebut server. Server menyimpan program web dan database untuk dapat diakses oleh *admin* atau *client* dari *browser*. *Website* dapat dibangun menggunakan program notepad/notepad++, Adobe Dreamweaver, atau Sublime.

Web server adalah sebuah *software* yang memberikan layanan berbasis data dan berfungsi menerima permintaan dari HTTP atau HTTPS pada *client* yang dikenal dan biasanya kita kenal dengan nama web browser dan untuk mengirimkan kembali yang hasilnya dalam bentuk beberapa halaman web dan pada umumnya akan berbentuk dokumen HTML. Dalam bentuk sederhana web server akan mengirim data HTML kepada permintaan web browser sehingga akan terlihat seperti pada umumnya yaitu sebuah tampilan *website* (Damayanti & Fradita, 2017).

Fungsi utama web server adalah untuk melakukan *transfer* berkas permintaan pengguna melalui protokol komunikasi yang telah ditentukan sedemikian rupa. Halaman web yang diminta terdiri dari berkas teks, video, gambar, *file* dan banyak lagi. Pemanfaatan web server berfungsi untuk mentransfer seluruh aspek pemberkasan dalam sebuah halaman web termasuk yang di dalam berupa teks, video, gambar atau banyak lagi (Damayanti & Fradita, 2017).

Beberapa jenis Web Server di antaranya adalah:

* + 1. Apache Web Server / The HTTP Web Server
    2. Apache Tomcat
    3. Microsoft windows Server 2008 IIS (Internet Information Services)
    4. Lighttpd
    5. Zeus Web Server
    6. Sun Java System Web Server
    7. NginX Web Server

Dari daftar di atas yang akan digunakan dalam penelitian ini ialah NginX karena arsitekturnya yang lebih ringan dan dapat meng-*handle* banyak *request* sekaligus.

### II.12.1 Web Server NginX

NginX (baca: engine x) adalah server HTTP dan Proxy dengan kode sumber terbuka yang bisa juga berfungsi sebagai *proxy* IMAP/POP3. Kode sumber nginx ditulis oleh seorang warga negara Rusia yang bernama Igor Sysoev pada tahun 2002 dan dirilis ke publik pada tahun 2004. Nginx terkenal karena stabil, memiliki tingkat performa tinggi dan minim mengonsumsi sumber daya. Sehingga sangat cocok dipasang di Mini PC yang mengonsumsi daya rendah (Barry Abrahamson, 2008).

Nginx juga memiliki fitur seperti *reverse proxy multiple protocols* (HTTP, Memcached, PHP‑FPM, SCGI, uwsgi), Stream HTTP video (FLV, HDS, HLS, MP4) serta HTTP/2 gateway.Beberapa situs terkenal yang menggunakan Nginx adalah WordPress, Fastmail, Ohloh, SourceForge, dan GitHub.

#### II.11.1.1 Keunggulan Web Server NginX

*Web Server NginX mempunyai kelebihan dari beberapa pertimbangan di*

atas (Loka Dwiartara, 2014):

1. Arsitektur NginX. Salah satu yang membuat Nginx menjadi sangat cepat adalah jenis arsitektur Nginx itu sendiri. Jika di bandingkan dengan apache yang *process based*, Nginx menjadi jauh lebih unggul karena *event-based* nya. Sehingga mampu memanfaatkan seminimal mungkin *thread* untuk memproses *request* dari *user*, sehingga akhirnya memori yang terpakai oleh Nginx menjadi minimal. Karena memori yang dipakai sangat kecil, maka hasilnya server menjadi ringan dan jauh-jauh lebih responsif (memiliki respon super cepat).

2. Performa Tinggi. Melihat dari arsitekturnya, NginX bisa meng-*handle* lebih banyak *request* daripada Apache. Hal ini dikarenakan NginX tidak menggunakan *thread* baru untuk setiap *request* yang diterima, melainkan bekerja dengan metode *event-based.* Jadi Thread baru akan dipekerjakan apabila ada sebuah *request* dari *user*.

3. Konfigurasi yang sangat mudah untuk dijalankan.

#### II.12.1.2 Fasilitas Web Server NginX

Fasilitas atau ciri khas dari *web server* NginX adalah *(Loka Dwiartara, 2014*) :

* + - Static file serving.
    - SSL/TLS support.
    - Virtual hosts.
    - Reverse proxying.
    - Load balancing.
    - Compression.
    - Access controls.
    - URL rewriting.
    - Custom logging.
    - Server-side includes.
    - FLV streaming

## II.13 Port Forwarding

Setiap perangkat yang terhubung ke jaringan memiliki identifikasi unik yaitu alamat IP. Alamat ini digunakan untuk mengidentifikasi dan berkomunikasi dengan perangkat. Misalnya jaringan rumah Anda. Anda mungkin akan memasang *router* yang terhubung ke jaringan Penyedia Layanan Internet (ISP). Router memiliki alamat *IP-IP* publik unik yang digunakan untuk mengidentifikasi perangkat dan berkomunikasi dengan internet. Anda menghubungkan berbagai perangkat ke *router* ini - baik menggunakan kabel *ethernet* ataupun menggunakan *wi-fi*. Perangkat ini memiliki alamat IP sendiri namun itu adalah alamat pribadi, artinya alamat ini digunakan untuk mengidentifikasi perangkat dalam jaringan rumah\lokal. Alamat IP ini berbeda dari alamat IP publik *router* dan perangkat di luar jaringan lokal tidak dapat berkomunikasi dengan perangkat yang hanya menggunakan alamat IP pribadi. Ketika salah satu perangkat ingin berkomunikasi dengan internet (perangkat di luar jaringan lokal), *router* mengganti alamat IP pribadi dengan alamat IP publiknya untuk lalu lintas keluar dan mengganti alamat publik dengan alamat IP pribadi khusus untuk semua lalu lintas masuk. Pemetaan ini disebut *Network Address Translation* (NAT).

Sama seperti perangkat yang membutuhkan identifikasi unik, setiap layanan perlu diidentifikasi untuk memungkinkan komunikasi. Nomor *port* membantu dalam mengidentifikasi layanan. Setiap kali layanan perlu berkomunikasi, nomor *port* membantu mengidentifikasi tujuan / sumber yang tepat pada perangkat dan membantu dalam transfer data yang sesuai. Sederhananya, alamat IP adalah alamat perangkat di jaringan sedangkan *port* adalah alamat layanan pada perangkat. Kombinasi alamat IP dan nomor *port* memungkinkan komunikasi dengan mengidentifikasi sumber / tujuan layanan tertentu pada perangkat tertentu. Sebagian besar layanan populer memiliki *port default* yang terkait dengannya, misalnya, *port* 80 untuk *HTTP*, *port* 443 untuk *https*, *port* 25 untuk *SMTP*, *port* 20 untuk *FTP*, dll.

*Port forwarding* adalah proses pengalihan komunikasi yang ditujukan ke satu alamat IP / kombinasi port ke alamat lain / kombinasi port. Tetapi mengapa kita membutuhkan port forwarding Port forwarding memungkinkan sebuah perangkat yang terhubung ke jaringan pribadi agar tersedia untuk diakses dari internet. Hal ini agar perangkat dari luar jaringan local rumah dapat diakses dari jaringan luar yang tidak terhubung sama sekali dengan jaringan lokal rumah.

### II.13.1 NGroK

Ngrok adalah proxy server untuk membuat / membuka jaringan private melalui *NAT* (*Network Address Translation*) atau *firewall* untuk menghubungkan localhost ke internet dengan tunnel yang aman. Ketika NGroK dimulai, itu akan menampilkan UI di terminal Anda dengan URL publik dari *tunnel* Anda dan informasi status dan metrik lainnya tentang koneksi yang dilakukan melalui terowongan Anda seperti gambar 2.4 berikut.



**Gambar 2.4** UI konsol ngrok

Dari gambar 2.4 diatas bisa dilihat pada tulisan “Forwarding” terdapat sebuah *link address* yang merupakan alamat yang dapat diakses dari luar jaringan lokal. Di sebelah kanan *link* tersebut terdapat *link address* local yang akan ditujunya. Jadi NGrok sebenarnya hanya meneruskan *link private* lokal dengan *tunnel* agar bisa diakses dari luar dengan *link public.*

# BAB III

# METODOLOGI PENELITIAN

## III.1 Tempat dan Waktu Penelitian

Penelitian ini dilaksanakan di Laboratorium Internet of Things dan Parallel Computing (IoT-PC), Departemen Teknik Informatika Fakultas Teknik, Universitas Hasanuddin. Kelurahan Romang Lompoa, Kecamatan Bontomarannu, Kabupaten Gowa. Sedangkan waktu penelitian dilaksanakan pada bulan Desember 2019 – April 2020.

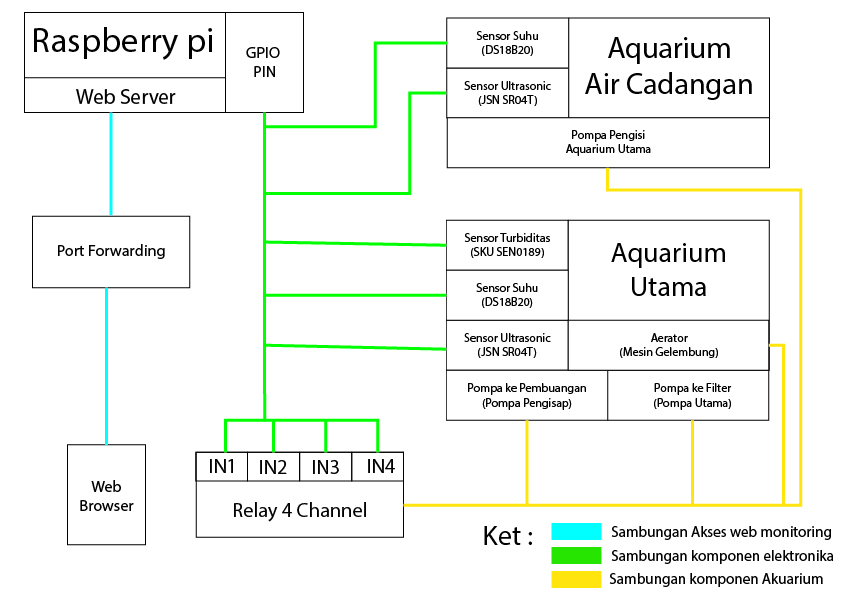
## III.2 Alat dan Bahan Penelitian

Alat yang digunakan pada penelitian ini yaitu solder listrik, obeng, lem tembak, perangkat komputer, perangkat lunak browser, database sqlite, Nginx, Python, sistem operasi Raspberry Pi , alat tulis, dan Multimeter.

Sedangkan bahan yang digunakan yaitu Raspberry Pi 4 Model B, Kabel Jumper, Micro SD, PCB, Timah, Terminal Blok, Terminal Kuningan, Raspberry Pi Adapter 5V 3A . Resistor 4.7K ohm ¼ Watt & 1K ohm ¼ Watt, Sensor Ultrasonik *Waterproof* JSN-SRT04, Sensor Suhu *Waterproof* DS18B20, ADC Converter ADS1115, Sensor turbiditas SKU-SEN018, Sensor Arus ACS712 5A, 4 *channel* Relay, Impura Board.

## III.3 Kriteria Desain

Sistem pemantauan dan perawatan akuarium berbasis *Internet of Things* ini dirancang untuk dapat memantau kondisi suhu , ketinggian, dan kekeruhan air di dalam akuarium dari jarak jauh melalui sebuah *website* yang dapat diakses oleh *user* walaupun sedang berada jauh dari lokasi.



**Gambar 3.1** Gambaran umum rangkaian sistem

Dari Gambar 3.1 gambaran umum alat dalam penelitian ini dapat dilihat di rangkaian sistem secara umum sebagai berikut:

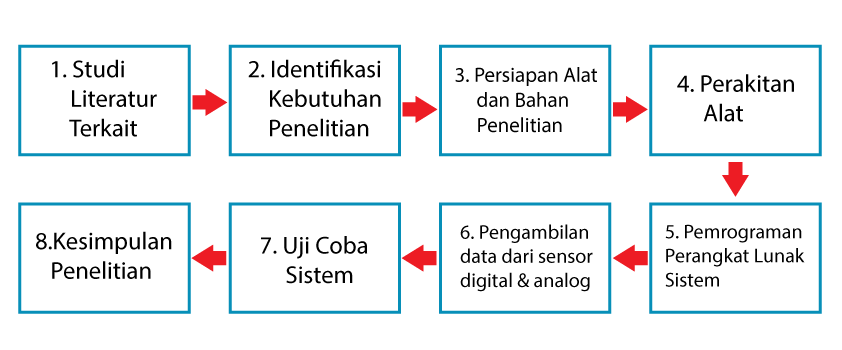
1. Ikan diberi makan oleh pemeliharanya.
2. Seiring berjalannya waktu, makanan yang dimasukkan ke dalam akuarium akan membuat air akuarium semakin keruh sehingga kekeruhan air ini akan dideteksi oleh sensor turbiditas yang diletakkan di dalam akuarium.
3. Sensor DS18B20 akan bertugas untuk memantau suhu air.
4. Sensor JSN- SRT04 akan bertugas untuk memantau ketinggian air.
5. Data nilai kekeruhan, suhu dan ketinggian air yang dideteksi oleh sensor kemudian akan diolah terlebih dahulu sebelum ditampilkan ke dalam sebuah halaman *web*  yang dapat menunjukkan kondisi akuarium berdasarkan nilai-nilai yang didapatkan oleh sensor.
6. Seluruh sensor dalam proyek ini terhubung langsung ke sebuah *Single Board Computer* bernama raspberry pi 4 Byang sekaligus bertindak sebagai web server yang akan menerima akses *request* http apabila pemelihara ikan ingin mengetahui kondisi akuarium melalui *web browser*.
7. Data terbaru yang masuk akan di tampilkan pada halaman web yang otomatis memuat ulang setiap 15 detik.

Dengan alur kerja di atas, data suhu, level, dan kekeruhan air di dalam akuarium akan ditampilkan pada halaman web sehingga dapat diakses dari jarak jauh. Pemantau dapat memantau data suhu, level, dan kekeruhan air di dalam akuarium yang dapat diakses pada PC ataupun g*adget smartphone* melalui W*eb Browser*.

## 

## III.4 Prosedur Penelitian

Tahapan pada penelitian ini sebagaimana ditunjukkan pada gambar 3.2.



**Gambar 3.2** Diagram tahapan penelitian

Tahapan secara garis besar dijelaskan sebagai berikut :

1. Pada tahapan studi literatur, pencarian penelitian dilakukan terkait rancang bangun sistem pemantauan dan perawatan akuarium berbasis *Internet of Things*. Pada tahap ini juga dilakukan pencarian dokumentasi hasil penelitian- penelitian sebelumnya yang terkait dengan topik penelitian.
2. Pada tahap ini, dilakukan penetapan berbagai kebutuhan penelitian dan disiapkan guna untuk menunjang penelitian dan perancangan sistem serta pembuatan aplikasi.
3. Pada tahap ini dilakukan pembelian alat dan bahan yang dibutuhkan untuk penelitian, seperti sensor turbiditas, sensor suhu, sensor ultrasonik, kabel jumper, dan lain-lain.
4. Pada tahap ini, dilakukan pembuatan dan perangkaian alat pada dua buah akuarium. Akuarium pertama bertindak sebagai akuarium yang di mana ikan hias tinggal, di dalam akuarium ini terdapat sensor suhu untuk memantau suhu , sensor ulrasonik untuk memantau ketinggian air, serta sensor turbiditas untuk memantau tingkat kekeruhan air serta terdapat dua buah pompa yaitu pompa utama yang menuju ke filter dan pompa pengisap yang menuju ke pembuangan air. Pada akuarium kedua terdapat sebuah pompa yang akan memompa air masuk ke akuarium utama dengan air bersih, juga terdapat sensor suhu untuk memantau suhu dan sensor ulrasonik untuk memantau ketinggian air. Adapun semua sensor pada sistem ini terhubung langsung ke raspberry pi yang sekaligus bertindak sebagai web server.
5. Pada tahap ini, pemrograman sistem perangkat lunak menggunakan bahasa pemrograman Python, Javascript, SQLite dengan tahapan pemrograman mulai dari pembacaan data sensor untuk ditampilkan pada halaman Web.
6. Pada tahap ini, data yang telah diterima oleh sensor akan diolah server agar data mentah dari sensor di konversi terlebih dahulu ke dalam bentuk *string* yang lebih mudah dibaca untuk kemudian ditampilkan pada halaman web.
7. Setelah melakukan tahapan-tahapan di atas, diperolehlah kesimpulan berdasarkan penelitian yang telah dilakukan.

## III.5 Tahap Persiapan

Tahap persiapan merupakan tahapan awal dalam melakukan penelitian. Pada tahap ini penulis melakukan studi literatur dengan mencari berbagai acuan baik melalui buku, jurnal, tugas akhir maupun artikel dengan narasumber yang jelas dan terpercaya dengan tujuan untuk melengkapi literatur mengenai penelitian ini. Penulis juga melakukan identifikasi masalah pada penelitian ini, membaca dan memahami kelebihan dan kekurangan dari penelitian yang sudah dilakukan sebelumnya. Serta penulis menyiapkan alat dan bahan yang diperlukan dalam penelitian ini untuk mempersiapkan menuju ke tahap selanjutnya.

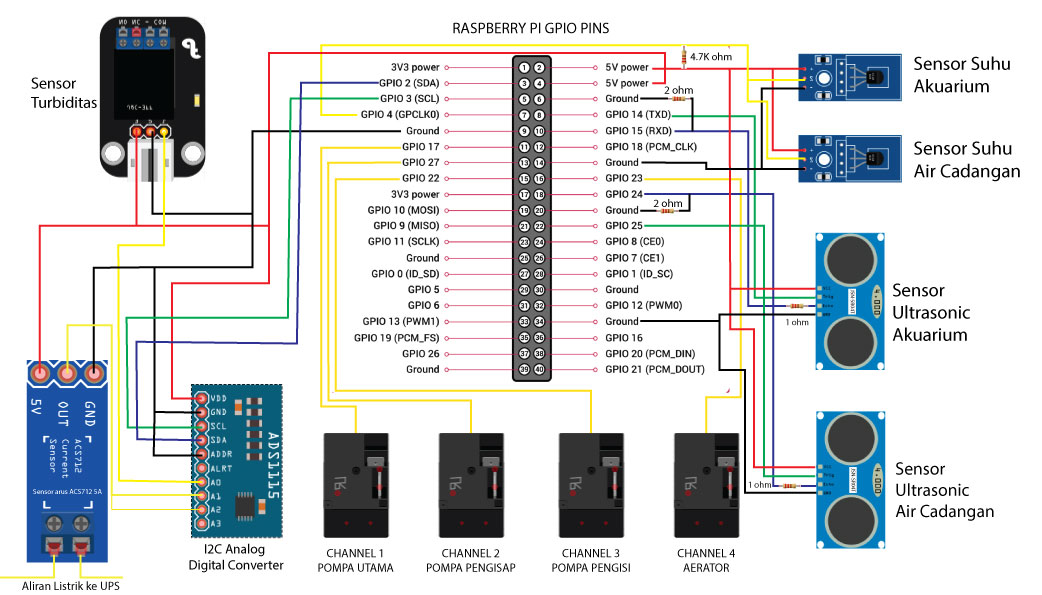
## III.6 Tahap Pembuatan Sistem

Tahap pembuatan sistem dibagi menjadi tiga tahap, yakni tahap perancangan sistem Hardware (Perangkat Keras), tahap perakitan / perwujudan alat, dan tahap pembuatan *software* (Perangkat Lunak). Berikut penjabaran dari masing-masing tahapan.

### III.6.1 Tahap Perancangan Sistem Hardware (Perangkat Keras).

Tahap perancangan sistem perangkat keras terdiri dari dua perancangan yaitu :

* 1. Perancangan rangkaian sistem. Skematik dari rangkaian sistem ini bisa dilihat pada gambar 3.3 berikut.



**Gambar 3.3** Rangkaian sistem

Perancangan skematik rangkaian sistem ini menggunakan aplikasi Fritzing dimana bertujuan untuk mempermudah saat proses perangkaian alat. Dalam menjalankan sistem, terdapat beberapa perangkat elektronik yang memiliki fungsinya masing-masing. Daftar komponen elektronik yang digunakan bisa dilihat pada tabel 3.1 berikut

**Tabel 3.1** Daftar komponen yang digunakan dalam sistem

|  |  |  |
| --- | --- | --- |
| **NO**. | **Nama Komponen** | **Kuantitas** |
| 1 | Raspberry Pi | 1 |
| 2 | Sensor DS18B20 | 2 |
| 3 | Sensor JSN-SR04T | 2 |
| 4 | Sensor SEN-0189 | 1 |
| 5 | Resistor 4.7K ohm ¼ Watt | 1 |
| 6 | Resistor 1K ohm ¼ Watt | 6 |
| 7 | Sensor ACS712 5A | 1 |

* 1. Perancangan rangkaian sistem akuarium penampungan air cadangan. Sistem akuarium penampungan air cadangan berfungsi sebagai tangki air yang akan mengisi akuarium utama dengan air bersih yang akan terintegrasi dengan sensor di akuarium utama untuk menyamakan kembali ketinggian air akuarium utama seperti sebelumnya setelah proses pembuangan air. Skematik rangkaian dapat dilihat pada gambar 3.3 .

#### III.6.1.1 Raspberry Pi 4 Model B

Raspberry Pi merupakan komputer kecil yang memiliki *input output* *digital port*. GPIO merupakan terminal *bi-directional* yang ditambahkan ke raspberry berjumlah 40 pin pada raspberry Pi 4 Model B. Pin GPIO (*General Purpose Input Output*) ini memiliki bagian-bagian yang mewakili 40 pin pada Raspberry Pi tersebut, antara lain : Power Supply (3.3Volt / 5Volt, 2 set), UART (*Universal asynchronous receiver/transmitter*, 1 set), SPI (*Serial Peripheral interface*), I2C (Inter-integrated Circuit). Tetapi pada penelitian kali ini *pin* yang digunakan ialah beberapa pin GPIO untuk mengambil data sensor ultrasonik, sensor suhu, relay, serta pin SDA/SCL sebagai media komunikasi I2C untuk modul ADC konverter untuk mengambil data dari sensor analog seperti sensor turbiditas. Pada sistem ini, Raspberry Pi 4 digunakan sebagai pengolah data yang diterima dari sensor dan memiliki fungsi sebagai Web Server.

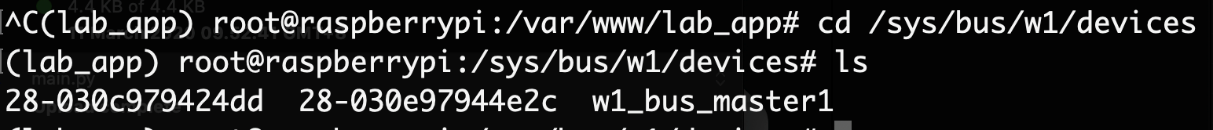
#### III.6.1.2 Sensor Suhu *Waterproof* DS18B20

Pada penelitian kali ini sensor suhu ini bertujuan untuk memantau suhu air di dalam akuarium. Di penelitian ini peneliti akan menggunakan dua buah sensor ini yang masing-masing akan ditempatkan di akuarium utama dan akuraium air cadangan. *Wiring* antara DS18B20dan Raspberry Pi dapat dilihat pada tabel 3.2 dan skematik penghubungannya dapat dilihat pada gambar 3.3 .

**Tabel 3.2** Wiring antara DS18B20 dan Respberry Pi

|  |  |  |
| --- | --- | --- |
| **DS18B20**  (Akuarium) | **DS18B20**  (Tank Air Cadangan) | **Raspberry Pi**  (Nomor Pin) |
| VCC | VCC | 2 |
| DATA | DATA | 7  (GPIO 4) |
| GND | GND | 9 |

Dari tabel 3.2 dapat dilihat bahwa kedua pin *data* sensor tersebut tersambung ke sebuah GPIO yang sama. Hal in dikarenakan sensor DS18B20 memiliki arsitektur 1*-Wire* yang memungkinkan penggunaan sensor ganda walaupun nilai *input-*nya akan masuk melalui satu buah pin data saja. Adapun cara untuk membedakan data dari masing-masing sensor ialah melalui id sensor itu sendiri. Dalam penelitian ini, *id* dari masing-masing sensor yang digunakan dengan awalan “28” dapat dilihat di gambar 3.4.



**Gambar 3.4** Id dari kedua buah sensor DS18B20

#### III.6.1.3 ADC Konverter ADS1115 16 Bit

Tujuan penggunaan modul ini dikarenakan raspberry pi tak memiliki modul ADC bawaan sedangkan peneliti ingin menggunakan dua buah sensor analog yaitu sensor turbiditas SEN-0189 dan sensor arus ACS712 5A. Nilai dari sensor ini tak bisa langsung diterima oleh *GPIO PINS* dari raspberry pi karena hanya bisa menerima sinyal digital. Modul ini sendiri menyediakan 4 buah pin analog untuk sensor-sensor digital yang ingin digunakan. *Wiring* antara ADS1115 , sensor analog (SEN-0189 & ACS712) dan raspberry pi dapat dilihat pada tabel 3.3 dan skematik penghubungannya dapat dilihat pada gambar 3.3 .

**Tabel 3.3** Wiring antara ADS1115 dan Sensor Analog

|  |  |  |  |
| --- | --- | --- | --- |
| Device | ADS1115 | | |
| GND | VDD | Analog Channel |
| SEN-0189 | GND | VCC | A0 |
| ACS712 | GND | VCC | A1 |

**Tabel 3.4** Wiring antara ADS1115 dan Raspberry pi

|  |  |
| --- | --- |
| **ADS1115** | **Raspberry pi**  **(Pin Number)** |
| VDD | 2 (5V) |
| GND | 6 (GND) |
| SCA | 3 (SCA) |
| SCL | 5 (SCL) |

#### III.6.1.3 Sensor Ultrasonik Waterproof JSN-SR04T

Pada penelitian ini sensor ultrasonik digunakan untuk memantau ketinggian air di dalam akuarium. Peneliti akan menggunakan dua buah sensor yang masing-masing akan ditempatkan di akuarium utama dan akuraium air cadangan. *Wiring* antara JSN-SR04T dan Raspberry Pi dapat dilihat pada tabel 3.5 dan skematik penghubungannya dapat dilihat pada gambar 3.3 .

**Tabel 3.5** Wiring antara JSN-SR04T dan Raspberry pi

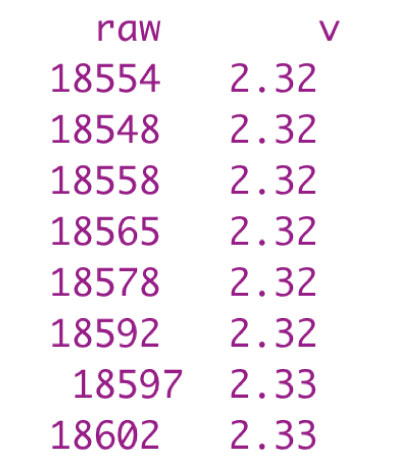
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Devices | Raspberry Pi GPIO Pins | | | |
| VCC | GND | TRIGGER | ECHO |
| JSN-SR04T  (Akuarium) | 5V | GND | GPIO 14 | GPIO 15 |
| JSN-SR04T  (Air Cadangan) | 5V | GND | GPIO 25 | GPIO 23 |

#### III.6.1.4 Sensor Turbiditas DFRobot SEN-0189

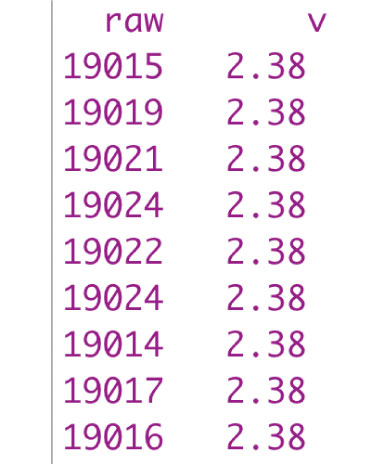
Sensor ini akan digunakan oleh peneliti untuk mengukur tingkat kekeruhan air di dalam akuarium. Sensor kekeruhan ini terdapat LED photodiode sebagai *transmitter* dan *photodiode* sebagai *receiver*. Sensor ini memanfaatkan cahaya yang dipancarkan pada LED yang kemudian hasil pemantulan cahaya akan dibaca oleh sensor, sehingga semakin tinggi tingkat kekeruhan air yang akan dideteksi maka tingkat pemantulan cahaya yang diterima akan semakin sedikit, dan sebaliknya. Pada pembelian sensor ini terdapat modul tambahan sebagai *op-amp* untuk menyediakan *output* berupa tegangan analog dan logika digital. Sehingga dapat digunakan sesuai kebutuhan pengaplikasian penggunaan sensor kekeruhan ini. *Wiring* antara Sensor Turbiditas DFRobot SEN-0189dan Raspberry Pi dapat dilihat pada tabel 3.3 dan skematik penghubungannya dapat dilihat pada gambar 3.3 .

#### III.6.1.5 Sensor Arus ACS712 5A

Dalam penelitian ini akan digunakan sebuah sensor arus. Yaitu untuk mendeteksi keberadaan arus listrik yang masuk ke dalam *UPS* untuk mendeteksi apakah sedang terjadi pemadaman listrik atau tidak.



**Gambar 3.5** Nilai Balik Sensor Arus Ketika Arus Listrik Terdeteksi



**Gambar 3.6** Nilai Balik Sensor Arus Ketika Arus Listrik Tidak Terdeteksi

Dari data hasil percobaan yang dilakukan peneliti diatas, dapat dibuat sebuah program yang akan mendeteksi apakah sedang terjadi pemadaman listrik atau tidak. Untuk mengetahui apakah sistem sedang dalam mode baterai atau tidak.

Namun karena raspberry pi yang menjadi pusat kendali dan komputasi dari sistem ini tak memiliki modul *ADC* bawaan yang bias meng-konversi sinyal analog menjadi sinyal digital, maka sensor ini tak bisa langsung disambungkan ke *GPIO PIN*  dari raspberry pi. Hal ini dikarenakan *GPIO PIN*  dari raspberry pi hanya dapat menerima sinyal digital. Oleh karena itu untuk menggunakan sensor ini dengan raspberry pi, diperlukan sebuah modul konverter ADC yang dapat mengkonversi nilai analog dari sensor ini menjadi nilai digital yang dapat dibaca oleh *GPIO PIN*  dari raspberry pi.

Modul yang digunakan dalam penelitian ini ialah ADS1115 yang menyediakan empat buah kanal *pin* analog untuk membaca nilai modul yang berupa sinyal analog seperti sensor arus ini. Adapun *wiring* antara sensor arus ACS712 5A dan ADS1115 dapat dilihat pada tabel 3.3 dan skematik penghubungannya dapat dilihat pada gambar 3.3 .

#### III.6.1.6 Relay 4 *channel* 5V

Pada penelitian kali ini peneliti menggunakan relay 4 *channel* sebagai aktuator untuk menyambung dan atau memutuskan aliran listrik dari komponen akuarium seperti pompa dan lampu akuarium.

Adapun sambungan setiap *channel* dalam *relay* adalah sebagai berikut :

1. Channel 1 : Pompa Utama (Akuarium Utama)
2. Channel 2 : Pompa Pengisap (Akuarium Utama)
3. Channel 3 : Pompa Pengisi (Tangki Air Cadangan)
4. Channel 4 : Lampu Akuarium

Setiap kanal dari *relay* ini akan terhubung ke GPIO PINS dari raspberry pi untuk mengontrol nilainya melalui program. Adapun *wiring* dari kanal *relay* ini dapat dilihat di tabel 3.6 sebagai berikut.

**Tabel 3.6** sambungan kanal relay dengan GPIO pins

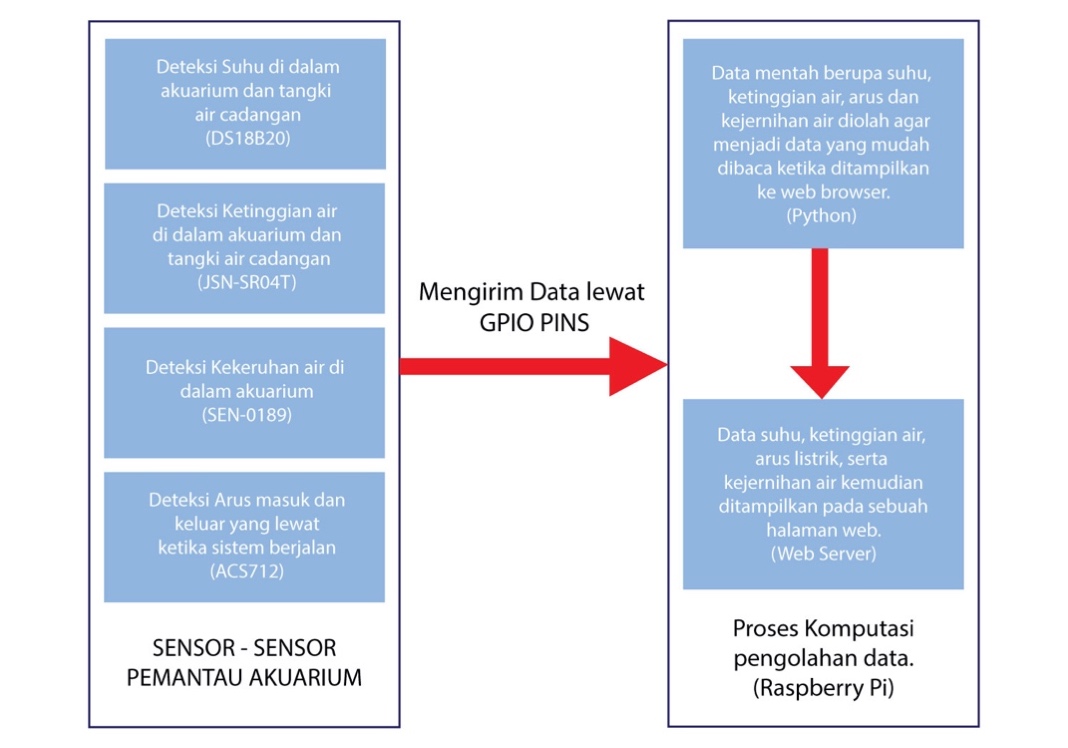
|  |  |
| --- | --- |
| **Kanal** | **GPIO PINS** |
| 1 | 17 |
| 2 | 27 |
| 3 | 22 |
| 4 | 23 |

### III.6.2 Tahap Perakitan / Perwujudan Alat

Tahapperakitan/ perwujudan alat ini merupakan tahap realisasi dari tahap perancangan alat. Tahap pembuatannya yakni yang pertama ialah pembuatan box elektronik tempat meletakkan stop kontak untuk setiap komponen akuarium, raspberry, relay, rangkaian kabel sensor ke raspberry (*breadboard*), dan sensor – sensor. Setelah box penampung dibuat, sensor ultrasonik kemudian diletakkan di akuarium utama dan akuarium air cadangan untuk memantau ketinggian air akuarium, begitu juga dengan sensor suhu untuk memantau suhu air kedua tangki akuarium. Namun dikarenakan jarak minimal yang dapat diukur oleh sensor JSN-SR04T adalah 20 cm, maka sensor ini dibuatkan terlebih dahulu tiang setinggi 25 cm dan diletakkan di atas akuarium agar dapat mengukur tinggi air akuarium saat penuh.

### III.6.3 Tahap Pembuatan Software (Perangkat Lunak)

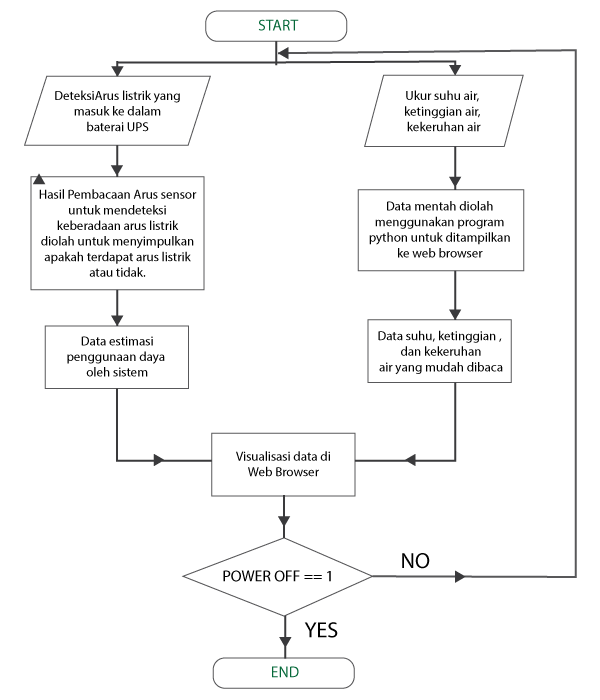
Tahap pembuatan perangkat lunak ini merupakan tahap pembuatan program untuk membuat rancangan perangkat keras yang telah dirakit dapat menjalankan fungsinya masing-masing di mana pembuatan program ini mengikuti blok diagram cara kerja alat yang bisa dilihat pada gambar 3.5 berikut.



**Gambar 3.7** Blok diagram cara kerja alat

Pada gambar 3.5, dapat dilihat alur program yang dibuat, di mana terdapat dua bagian yaitu pertama sensor-sensor yang diletakkan di akuarium untuk memantau kondisi akuarium. Sensor-sensor ini dihubungkan langsung ke raspberry pi melalui *GPIO pins* menggunakan kabel jumper.

Perlu diketahui bahwa data-data yang diterima oleh raspberry pi dari sensor bukanlah data yang bisa langsung ditampilkan di dalam web browser dikarenakan bentuk datanya yang sulit dibaca. Hal ini menyebabkan data yang diterima oleh sensor harus di *parsing* terlebih dahulu agar berubah menjadi bentuk yang lebih mudah dibaca oleh *user* menggunakan bahasa pemrograman python. Adapun alasan peneliti menggunakan python dikarenakan python adalah salah satu bahasa pemrograman yang kompatibel dengan sistem operasi raspberry pi yaitu *raspbian buster lite* dan merupakan salah satu bahasa yang paling banyak digunakan dalam proses pengolahan data selain Ruby*.*

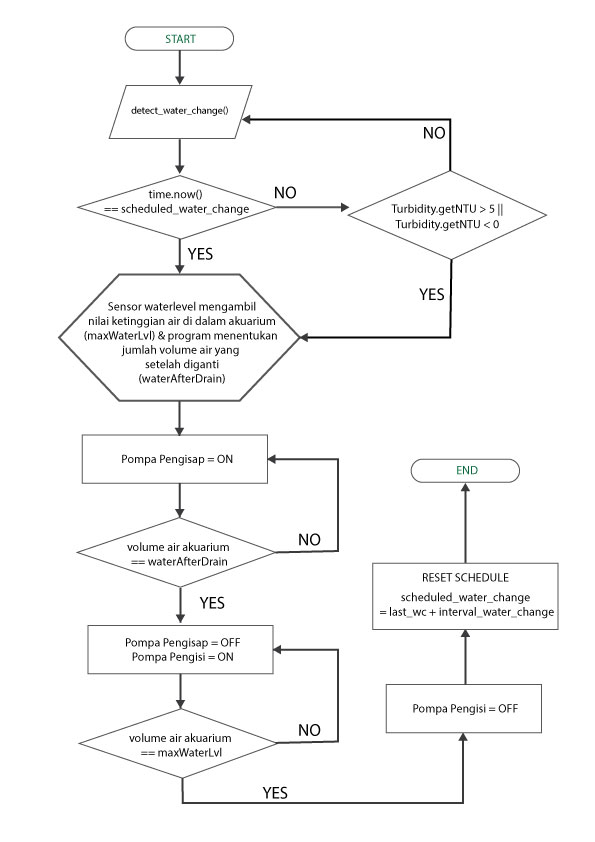


**Gambar 3.8** Flowchart sistem pemantauan akuarium

#### III.6.3.1 Sistem Penggantian air otomatis

Otomatisasi penggantian air dalam penelitian ini terbagi atas dua buah metode, yang pertama ialah metode berdasarkan waktu, dan yang kedua adalah berdasarkan kekeruhan air yang ditunjukkan oleh sensor turbiditas. Umumnya proses penggantian air tergantung kepada berbagai banyak faktor namun kebanyakan pemelihara ikan hias memiliki kebiasaan mengganti air berdasarkan interval waktu yang sama tiap minggunya dan atau menggantinya saat air sudah mulai terlihat keruh.

Untuk metode berdasarkan waktu, peneliti akan memprogram modul penggantian air untuk dieksekusi tiap minggu. Sedangkan untuk metode berdasarkan kekeruhan, peneliti akan memprogram modul penggantian air untuk dieksekusi apabila nilai kekeruhan air yang dideteksi oleh sensor turbiditas telah memenuhi nilai air keruh yang sebelumnya telah diketahui sebelumnya. Adapun *flowchart* dari metode penggantian air dapat dilihat di gambar 3.9.



**Gambar 3.9***Flowchart* Sistem penggantian air otomatis

#### III.6.3.2 Menghitung Penggunaan Daya

Dalam sistem yang dibuat ini, akan dibuat fitur dimana *user* akan bisa melihat berapa lama lagi sistem ini bisa berjalan dengan *UPS* apabila terjadi pemadaman listrik. Dalam penelitian kali ini digunakan sebuah *UPS* dengan kapasitas 12 V/50 Ah.

Perhitungan berapa lama UPS dapat mem*-backup* beban :

Rumus dasar :

I = Kuat Arus (Ampere) yang dideteksi oleh sensor arus

P = Daya (Watt)

V = Tegangan (Volt)

P = V x I  
V = P/I  
I = P/V

P = 25 W(pompa) + 50W (Heater) = 75 W

V = 12 V   
I = 75 / 12 = 6,25 Ampere

Maka didapat :  
Waktu pemakaian = 50 Ah/6,25 A

= 8 jam - defisiensi baterai sebesar 20 % (nilai perkiraan)  
= 8 jam – 1,6 jam  
= 6,4 Jam ( 6 Jam 24 Menit)

Jadi dari rumus dasar di atas, dapat diketahui bahwa lama ketahanan *UPS* ditentukan oleh besarnya kapasitas baterai dan berapa watt beban.

#### III.6.3.3 Visualisasi Data ke Halaman Web

Visualisasi data yang diterima raspberry pi untuk ditampilkan halaman web dapat dilihat pada gambar 3.6 . Dimana proses pertama adalah mengolah (*parse)* data yang ber-*tipe* raw ke variabel ber-tipe *string* agar dapat dilihat dengan lebih mudah. Data yang telah diolah akan di-*render* ke halaman web dengan laman *HTML* di widget masing masing yang telah disediakan. Data terakhir yang masuk akan ditampilkan pada halaman web secara *realtime*. Halaman web akan otomatis memuat ulang setiap 10 detik. Metode penampilan data yang telah diolah untuk ditampilkan ke laman web akan dijelaskan lebih lanjut di bab IV.

# BAB IV HASIL DAN PEMBAHASAN

## IV.1 Hasil Rancang Bangun Alat

Setelah semua desain rancangan dan komponen penyusunnya telah selesai dibuat maka tahap selanjutnya adalah perakitan komponen-komponen elektronik, rangkaian kabel sensor-sensor , serta peletakan sensor-sensor di akuarium menjadi satu kesatuan tiap sistem pada box komponen elektronik tempat raspberry pi, rangkaian sensor, *relay*, dan stop kontak yang telah dibuat. Sehingga dapat menjalankan fungsi sebagaimana blok diagram pada gambar 3.1.



**Gambar 4.1** Box prototype tempat penyimpanan komponen elektronika

Pada gambar 4.1, Box *prototype* ini berfungsi sebagai tempat atau wadah untuk rangkaian elektronika dari sensor-sensor yang digunakan. Sehingga dapat berfungsi sebagaimana mestinya.



**Gambar4.2** Penempatan sensor ultrasonik untuk mengukur ketinggian air.

Pada gambar 4.2 dapat dilihat merupakan penempatan sensor ultrasonik di atas akuarium yang telah terpasang di sebuah tiang yang memiliki tinggi 25 cm. Alasan peneliti menempatkan sensor ini di tiang ialah karena jangkauan jarak yang dapat dideteksi oleh sensor tersebut ialah minimal 20 cm dan maksimal 600 cm.



**Gambar 4.3**Penempatan sensor suhu di air akuarium cadangan

Pada gambar 4.3 dapat dilihat sensor suhu DS18B20 yang dicelupkan ke dalam air akuarium. Hal ini bertujuan untuk mengukur suhu air akuarium untuk nantinya hasil pengukuran tersebut ditampilkan di sebuah halaman web.



**Gambar 4.4**Penempatan sensor turbiditas di filter air akuarium

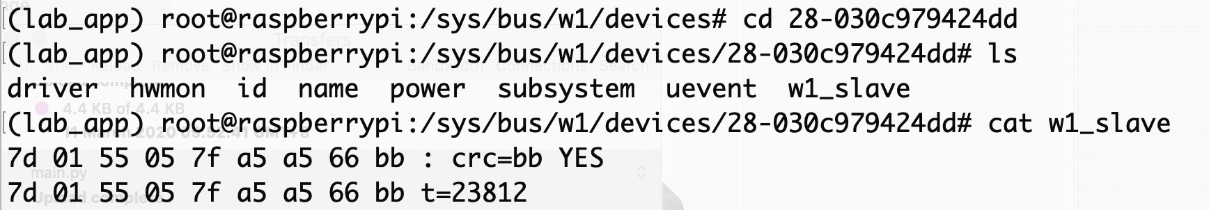
Pada gambar 4.4 dapat dilihat sensor turbiditas SEN-0189 yang dicelupkan ke dalam filter air akuarium. Hal ini bertujuan untuk mengukur tingkat kekeruhan air akuarium untuk nantinya dijadikan sebagai variabel dalam mempertimbangkan keputusan untuk mengeksekusi protokol penggantian air dengan metode kekeruhan air yang pada subbab selanjutnya akan dibahas lebih dalam.

## IV.2 Pengujian Sistem

Pengujian yang dilakukan meliputi pengujian sistem pembacaan suhu, ketinggian air, dan kekeruhan air yang digunakan oleh sistem ini selama beroperasi, kemudian pengujian program otomatisasi penggantian air serta pengujian visualisasi data pada halaman *web*. Pengujian ini bertujuan untuk mengetahui sistem dapat berjalan sesuai dengan *listing* program yang telah diprogramkan.

### IV.2.1 Pengujian Sistem Pembacaan Suhu

Pengujian sistem pembacaan suhu dan kelembapan ini menggunakan sensor DS18B20.



**Gambar 4.5**Tampilan Data hasil pembacaan suhu air akuarium pada terminal

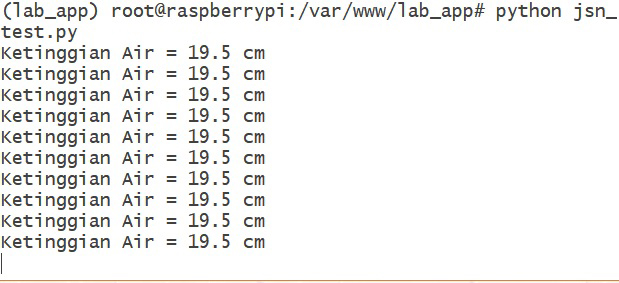
Pada gambar 4.5 menunjukkan pembacaan data suhu dari sensor DS18B20 pada akuarium utama dimana “t=23812” merupakan label dari *Temperature* dengan data suhu 23.8120C yang masih berbentuk *raw* dan tentu saja harus di *parse* terlebih dahulu ke dalam bentuk *stirng* sederhana agar bisa dengan mudah dibaca oleh *user* dari halaman *web.*

|  |
| --- |
| #SCRIPT AquariumAutomation.DSTemp.py  def read\_temp\_raw(device\_file):  f = open(device\_file, 'r')  lines = f.readlines()  f.close()  return lines    def read\_temp(device\_file):  lines = read\_temp\_raw(device\_file)  while lines[0].strip()[-3:] != 'YES':  time.sleep(0.2)  lines = read\_temp\_raw(device\_file)  equals\_pos = lines[1].find('t=')  if equals\_pos != -1:  temp\_string = lines[1][equals\_pos+2:]  temp\_c = float(temp\_string) / 1000.0 # Get the tempereture in celcius  return str("%.2f" % temp\_c) |

Dari skrip diatas terdapat baris kode “ equals\_pos = lines[1].find('t=') ” . Kode ini berfungsi untuk mengambil nilai yang tertera setelah “t=” yang merupakan nilai raw dari hasil pembacaan sensor suhu. Kemudian di bawahnya terdapat baris kode “temp\_c = float(temp\_string) / 1000.0” untuk mengubah nilai mentah dari hasil pembacaan sensor tersebut ke dalam satuan Celsius dan terakhir baris kode “ return str("%.2f" % temp\_c)” untuk mendapatkan nilai suhu dengan format dua angka di belakang koma agar mudah dibaca ketika ditampilkan di halaman web.

### IV.2.2 Pengujian Sistem Pembacaan Ketinggian Air

Pengujian pada sistem pembacaan ketinggian air ini dilakukan dengan cara menempatkan sensor ultrasonik di atas akuarium dengan tiang setinggi 25 cm agar pembacaan lebih akurat seperti pada gambar 4.2 .



**Gambar 4.6**Hasil Pembacaan ketinggian air akuarium

Pada gambar 4.6 dapat dilihat bahwa hasil pembacaan ketinggian air akuarium saat itu ialah 19,5 cm. Adapun untuk mengetahui ketinggian air, peneliti menggunakan rumus sebagai berikut.

X = Jarak terukur sensor ke dasar akuarium ketika kosong

Y = Jarak terukur sensor ke permukaan air akuarium terisi

**Ketinggian air = X - Y**

### IV.2.3 Pengujian Sistem Pembacaan Kekeruhan Air

Pengujian pada sistem pembacaan kekeruhan air ini dilakukan dengan cara mencelupkan sensor kekeruhan ke dalam filter air akuarium. Pencelupan sensor ini bertujuan untuk mendeteksi kekeruhan pada air akuarium seiring berjalannya waktu untuk mengambil keputusan apakah ini waktunya untuk mengganti air atau tidak.

Penjelasan lebih lanjut terhadap pengujian sistem pembacaan kekeruhan air akan dijelaskan pada subbab berikutnya yang berjudul Pengujian Sistem Penggantian Air Otomatis Metode Kekeruhan.

### IV.2.4 Pengujian Sistem Penggantian Air Otomatis

Pengujian pada sistem penggantian air otomatis ini dilakukan dengan dua metode yang berbeda. Yaitu metode mengganti air berdasarkan waktu (sesuai jadwal yang telah diatur sebelumnya) dan berdasarkan tingkat kekeruhan air. Berikut adalah skrip AutoWC.py yang merupakan program python yang akan mengatur logika dalam proses penggantian air secara otomatis dengan pengambilan keputusan inisiasi pergantian air berdasarkan kekeruhan air dan berdasarkan jadwal.

def start\_water\_changing():

while isWaterChangeCompleted == False:

time.sleep(2)

if(isTankDrained == False):

drain\_half\_tank()

else :

fill\_full\_tank()

if(isWaterChangeCompleted == True):

print("Water Change completed")

break

def drain\_half\_tank():

# half = maxWater/2 #kode asli

if (float(JSNDistance.measureTank(GPIO\_TRIGGER\_MAIN,

GPIO\_ECHO\_MAIN,mainTankDistance)) > float(volumeAfterDrain)):

GPIO.output(pengisap, GPIO.LOW) #Nyalakan pompa pengisap

time.sleep(4)

print("penyedotan sedang berlangsung...")

print("Target volume air :")

print(volumeAfterDrain)

print("Ketinggian Air : ")

print(JSNDistance.measureTank(GPIO\_TRIGGER\_MAIN,

GPIO\_ECHO\_MAIN,mainTankDistance))

print("\n\n")

else :

GPIO.output(pengisap, GPIO.HIGH) #Matikan pompa pengisap

globals()['isTankDrained'] = True

print("penyedotan telah selesai!\n\n")

print("\n\n")

time.sleep(1)

def fill\_full\_tank():

if(float(JSNDistance.measureTank(GPIO\_TRIGGER\_MAIN,

GPIO\_ECHO\_MAIN,mainTankDistance)) < float(maxWater)):

GPIO.output(pengisi, GPIO.LOW) #Nyalakan pompa pengisi

time.sleep(2)

print("pengisian sedang berlangsung...\n")

print("Ketinggian Air : ")

print(JSNDistance.measureTank(GPIO\_TRIGGER\_MAIN,

GPIO\_ECHO\_MAIN,mainTankDistance))

print(" Target volume air :")

print(maxWater)

print("\n\n")

else :

GPIO.output(pengisi, GPIO.HIGH) #Matikan pompa pengisi

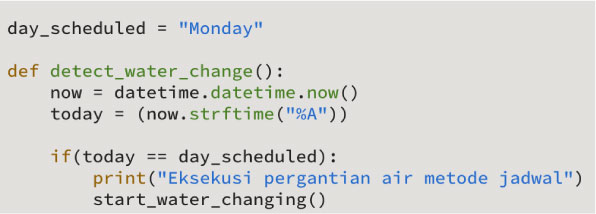
print("mematikan pompa pengisi...")

globals()['isWaterChangeCompleted'] = True

Kodeprogram diatas digunakan untuk mendeteksi ketinggian air di akuarium utama selama proses penggantian air guna mengambil keputusan kapan menyalakan dan mematikan pompa pengisap dan pompa pengisi agar proses penggantian air tetap mempertahankan volume air pada ketinggian yang ideal.

#### IV.2.4.1 Pengujian Sistem Penggantian Air Otomatis Metode Waktu

Pengujian dengan metode waktu itu ialah menguji apakah sistem ini akan mengeksekusi perintah untuk melakukan proses penggantian air pada jadwal yang telah diatur sebelumnya.



**Gambar 4.7** Jadwal disetel untuk mengganti air tiap hari senin



**Gambar 4.8**Kondisi akuarium penuh saat air belum diganti

****

**Gambar 4.9** Kondisi akuarium saat air sedang diganti



**Gambar 4.10**Kondisi akuarium saat proses penggantian air telah selesai dieksekusi

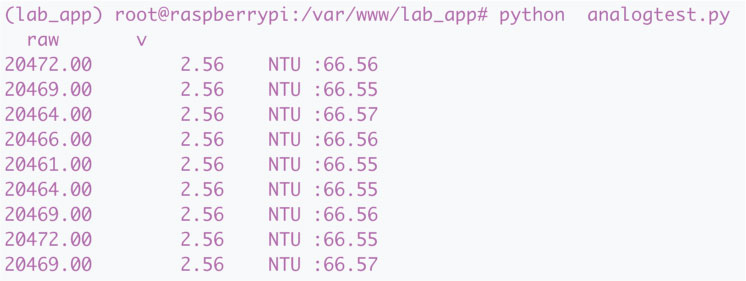
Dapat dilihat dari gambar 4.7 sampai gambar 4.10 bahwa prosedur penggantian air akuarium telah berhasil berjalan dengan akurasi ketepatan pembacaan sensor ultrasonik terhadap ketinggian air sebenarnya mencapai 97,62%. Hal ini menunjukkan bahwa metode penggantian air dengan metode jadwal telah berhasil dilaksanakan.

|  |
| --- |
| def detect\_water\_change():  now = datetime.datetime.now()  today = (now.strftime("%A"))  if(today == day\_scheduled): # True  print("Eksekusi pergantian air metode jadwal")  start\_water\_changing()  elif (SENTurbidity.check\_turbidity\_values == True):  print("Eksekusi pergantian air metode kekeruhan")  start\_water\_changing()  else:  print("Air aman") |

Kode program di atas merupakan kode program yang digunakan untuk mengeksekusi *method* untuk mengganti air. Adapun isi dari *method* yaitu untuk menyalakan pompa pengisap di akuarium utama hingga ketinggian air berkurang sesuai dengan jumlah yang diatur. Kemudian setelah itu program akan menyalakan pompa pengisi untuk mengisi kembali air di dalam akuarium agar ketinggian air di dalam akuarium kembali seperti semula.

#### IV.2.4.2 Pengujian Sistem Penggantian Air Otomatis Metode Kekeruhan Air

Pengujian dengan metode kekeruhan air ialah menguji apakah sistem ini akan mengeksekusi perintah untuk melakukan proses penggantian air pada saat nilai kekeruhan air di dalam akuarium telah mencapai nilai air keruh yang sudah diatur sebelumnya.

****

**Gambar 4.11**Nilai tegangan air keruh

**Gambar 4.12** Air akuarium keruh



**Gambar 4.13** Kondisi akuarium saat air sedang diganti



**Gambar 4.14**Kondisi akuarium saat proses penggantian air telah selesai dieksekusi

Dapat dilihat dari gambar 4.12 sampai gambar 4.14 bahwa prosedur penggantian air akuarium telah berhasil berjalan dengan akurasi ketepatan pembacaan sensor ultrasonik terhadap ketinggian air sebenarnya mencapai 95,26%.. Hal ini menunjukkan bahwa metode penggantian air dengan metode nilai kekeruhan air telah berhasil dilaksanakan.

Kode program yang digunakan merupakan kode program yang sama dalam metode kekeruhan. Skrip yang menginisiasi proses penggantian air ini dieksekusi dengan *cron job* agar *main task* dari sistem ini cukup berfokus untuk mengerjakan *task* yang berhubungan dengan *request* ke *Web Server* agar tidak mengganggu performa dalam proses penampilan data ke halaman web secara *real time.*

### IV.2.5 Pengujian Sistem Pemantauan Data Pada Web Browser

Pengujian visualisasi data pada *web server*  berupa pengujian penampilan data pada halaman *web*. Data yang diterima oleh sensor akan di-*parse*  terlebih dahulu sebelum ditampilkan ke halaman web, proses tersebut untuk menghapus karakter yang tidak penting yang ikut dalam data hasil pembacaan sensor. Berikut adalah kode program untuk menampilkan data ke halaman web.

# Mengambil nilai ketinggian air akuarium

var\_mainTankWaterLvl = JSNDistance.measureTank(

GPIO\_TRIGGER\_MAIN,

GPIO\_ECHO\_MAIN,

mainTankDistance)

# Mengambil nilai ketinggian air cadangan

var\_secondaryTankWaterLvl = JSNDistance.measureTank(

GPIO\_TRIGGER\_SECONDARY,

GPIO\_ECHO\_SECONDARY,

secondTankDistance)

# Mengambil nilai ketinggian air akuarium

var\_mainTankTemperature = DSTemp.read\_temp(device\_file1)\

# Mengambil nilai ketinggian air cadangan

var\_secondaryTankTemperature = DSTemp.read\_temp(device\_file2)

return render\_template('home.html',

mainTankWaterLvl = var\_mainTankWaterLvl,

secondaryTankWaterLvl = var\_secondaryTankWaterLvl ,

mainTankTemperature = var\_mainTankTemperature,

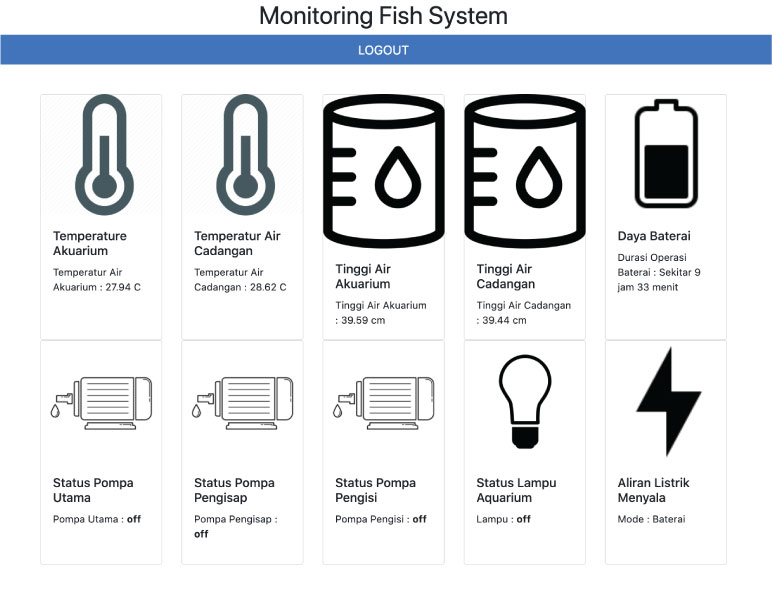
secondaryTankTemperature = var\_secondaryTankTemperature ,

\*\*templateData, hour=str("%.0f" % hour), minutes=str("%.0f" % minutes))

Data yang didapat setelah memanggil *library* masing-masing sensor kemudian di masukkan ke dalam argumen dalam fungsi  *render\_template()* yang merupakan fungsi bawaan dari *flask.render\_template().* Fungsi ini penulis meletakkan delapan argumen dengan urutan sebagai berikut :

* 1. Nama *file .html* yang akan ditampilkan
  2. Nilai ketinggian air akuarium (mainTankWaterLvl)
  3. Nilai ketinggian air cadangan (secondaryTankWaterLvl)
  4. Nilai suhu air akuarium (mainTankTemperature)
  5. Nilai suhu air cadangan (secondaryTankTemperature)
  6. Status pompa air dan lampu Akuarium (\*\*templateData)
  7. Estimasi Durasi baterai apabila mati lampu satuan jam (hour)
  8. Estimasi Durasi baterai apabila mati lampu satuan menit (minutes)

Dapat dilihat dari kedelapan argumen di atas, semua nilai dari sensor dan aktuator di-*render* ke dalam sebuah halaman web berformat *.html*. Hal ini agar dari sisi *client* dapat melihat nilai sensor dari *web server* pada sebuah halaman web melalui *web browser* apabila melakukan *request HTTP* pada web server pada laman *home* seperti berikut.



**Gambar 4.15** Pemantauan Data pada halaman web

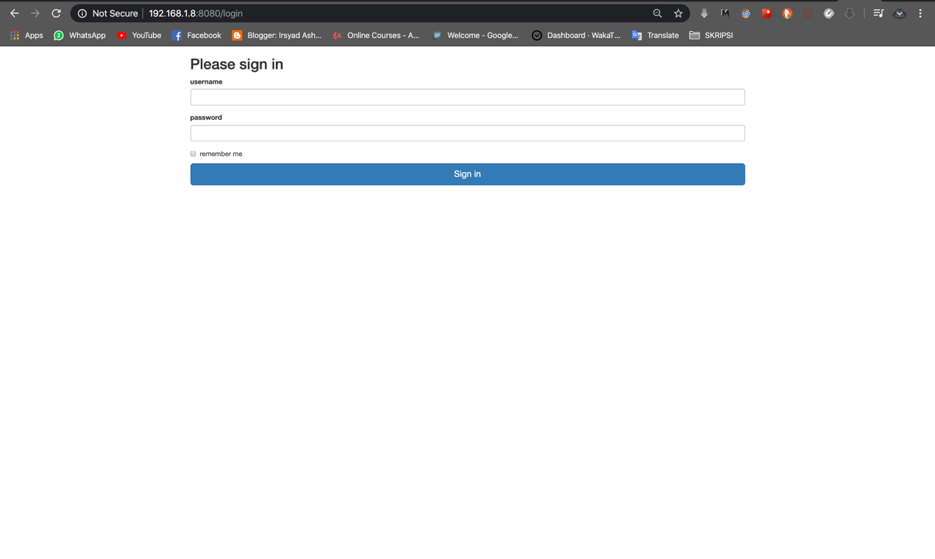
Pada gambar 4.15 dapat dilihat data terakhir ditampilkan pada halaman web yang akan di muat ulang setiap 15 detik. Data yang ditampilkan berupa data suhu air akuarium, suhu air cadangan, tinggi air akuarium, tinggi air cadangan, status nyala pompa utama, status nyala pompa pengisap, status nyala pompa pengisi dan status nyala lampu akuarium.

## IV.3 Hasil Pengamatan dan Data

Setelah dilakukan pengujian sistem, peneliti melakukan pengamatan pada sistem-sistem serta data yang dihasilkan.

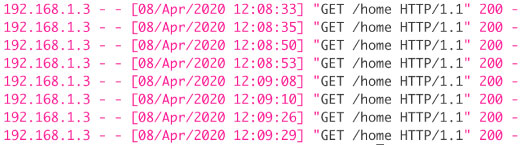
### IV.3.1 Halaman Web

Halaman web digunakan sebagai visualisasi dari implementasi *Internet of Things*. Di mana data suhu, ketinggian air, kontrol relay, dan kekeruhan air dapat dipantau di laptop ataupun *smartphone* dari lokasi manapun. Berikut tampilan halaman web-nya dengan halaman awal merupakan halaman *login* yang dapat dilihat pada gambar 4.16.



**Gambar 4.16** Halaman login

Pada gambar 4.16 dapat dilihat bahwa sebelum masuk ke *dashboard* pemantauan akuarium (*home*) , *user* diharapkan untuk *login* terlebih dahulu, hal ini bertujuan agar tidak semua orang dapat mengakses dan melihat kondisi akuarium kita melalui internet, melainkan hanya untuk orang yang telah ter-otorisasi



**Gambar 4.17***User* *request* ke server setiap 15 detik

Setelah *user* memasukkan *username* dan *password*, *user* akan dialihkan menuju laman *home* dimana seluruh kondisi akuarium dapat dipantau secara *remote* melalui perangkat apapun yang memiliki koneksi internet dan *web browser.* Gambar 4.17 menunjukkan 8 baris *log* *request* dengan kode 200 yang berarti *request* dari *client* telah dipenuhi oleh *web server* yang artinya *user* telah berhasil memuat laman *home* yang akan dimuat ulang setiap 15 detik untuk mendapatkan data terbaru yang didapatkan oleh sensor dengan tingkat keberhasilan 100%.

### IV.3.2 Penggantian Air Otomatis

Dalam penelitian ini dilakukan dua kali uji coba mengganti air. Yaitu dengan metode deteksi kekeruhan dan penjadwalan rutin. Hasil penggantian air otomatis tersebut menghasilkan data ketinggian air setelah melakukan proses penggantian air sebagai berikut.

**

**Tabel 4.1** Data Akurasi pembacaan sensor ketinggian air

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NO | Metode  Percobaan | Ketinggian Air Sebenarnya (cm) | Ketinggian hasil pembacaan sensor (cm) | **Error** | Akurasi ketinggian air sebenarnya terhadap pembacaan sensor |
| 1 | Metode Kekeruhan | 37,1 | 38,86 | 4.74% | 95,26 % |
| 2 | Metode Penjadwalan | 37,8 | 38,70 | 2.38% | 97.62 % |
| Akurasi rata-rata | | | | 96,44 % | |

Dari hasil kedua pengujian diatas , maka diperoleh rata-rata akurasi sensor JSN-SR04T yaitu 96,44 % sehingga dapat disimpulkan bahwa kinerja dari sensor sudah bekerja secara efektif.

# BAB V

# PENUTUP

Bab ini berisi tentang kesimpulan yang didapatkan berdasarkan hasil penelitian yang telah dilakukan serta saran-saran untuk pengembangan lebih lanjut.

## V.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan dalam pengujian sistem akuarium terotomatisasi berbasis *Internet of Things*, maka dapat disimpulkan bahwa:

* + 1. Berdasarkan hasil penelitian yang dirancang dapat diambil kesimpulan bahwa penggunaan sensor dengan Mini PC berbasis *Internet of Things (IoT)* dapat membantu untuk memantau kualitas air dan penggantian air akuarium menjadi lebih efisien. Penggunaan Raspberry pi sebagai mikrokontroler sekaligus *web server* dapat mengurangi waktu perawatan akuarium sekaligus mengurangi beban perawatan akuarium ikan hias. Bahkan dengan menjadikannya sebagai web server untuk situs pemantauan, pemelihara ikan hias dapat memantau kondisi akuariumnya dari jarak jauh. Pada pengujian sistem penggantian air otomatis didapatkan rata-rata akurasi sensor ultrasonik setelah penggantian air sebesar 96,44% pada 2 kali percobaan yaitu penggantian air berdasarkan waktu dan penggantian air berdasarkan tingkat kekeruhan air.
    2. Sistem pemantauan dan pemeliharaan akuarium terotomatisasi mengimplementasikan *Internet of Things*  dalam memantau suhu, ketinggian air, dan kekeruhan air pada halaman *web* sehingga berhasil dipantau dari jarak jauh. Hal tersebut dapat terealisasikan dikarenakan pada sistem ini digunakan mini pc yang bertindak sebagai web server yang terhubung ke jaringan internet secara nirkabel. Data dari sensor – sensor yang terhubung langsung ke mini pc melalui *GPIO PINS* kemudian di-*render* ke sebuah halaman web berformat *.html* untuk menampilkan data tersebut agar bisa dipantau secara *remote* oleh pengguna.

## V.2 Saran

Sehubungan dengan selesainya proses pembuatan skripsi ini, penulis bermaksud menyampaikan beberapa saran kepada para pembaca yakni:

* + 1. Peneliti berharap sistem pemantauan dan otomatisasi penggantian air ini dapat dikembangkan lagi dengan menambahkan fitur pengaturan otomatis suhu sehingga dapat membantu para pemelihara ikan hias dalam mengatur kondisi air secara otomatis.
    2. Pada proses pendeteksian ketinggian air, akurasi deteksi menggunakan sinyal ultrasonik belum mencapai 100%. Hal ini dikarenakan pemantulan gelombang ultrasonik dipengaruhi permukaan air yang bergelombang apabila ikan bergerak di dekatnya.
    3. Masih terdapat banyak variabel-variabel pada pemeliharaan ikan hias yang dapat dimanipulasi. Dimana peneliti berharap kedepannya variabel-variabel tersebut dapat di manipulasi sehingga tercipta satu sistem pemantauan dan pemeliharaan akuarium menggunakan teknologi-teknologi modern.

# 

# DAFTAR PUSTAKA

Asmanditya Hibatullah,(2019), Smart Aquarium Berbasis IOT. Surakarta: Inoversitas Muhammadiyah Surakarta.

Thiyraash Al David (2017), Aquarium Monitoring System. Kuala Lumpur : Tunku Abdul Rahman University.

M.Mahendran , G. Sivakannu , Sriraman Balaji (2017), Implementation of Smart Farm Monitoring Using IOT. Thiruchirapalli : Saranathan College Of Engineering

Muhammad Fachrial Yuni Yunizar Yunus, S. T. (2019). Rancang Sistem Rumah Walet Cerdas Berbasis Internet of Things. Makassar: Universitas Hasanuddin.

Fish Lore, (2013), Freshwater Aquarium Book.[Online]

Available At fishlore.com :

https://www.fishlore.com/freshwater-aquarium-book.pdf

Larry McGee, (2018), Aquarium Maintenance Guide.[Online]

Available At aquariummaintenancebusiness.com:

http://aquariummaintenancebusiness.com/wp-content/uploads/2019/07/Aquarium-Maintenance-Guide-eBook.pdf

Wpnsmith. (25. 7 2018). Raspberry Pi Temperature & Humidity Network Monitor : 11 Steps. [Online]

Available At Instructables.com: [https://www.instructables.com/id/Raspberry-Pi-Temperature-Humidity- Network-Monitor/](https://www.instructables.com/id/Raspberry-Pi-Temperature-Humidity-%20Network-Monitor/)

Dr. Peter Dalmaris. (25. 10 2019). Raspberry Pi Full Stack Raspbian. [Online]

Available At udemy.com:

<https://www.udemy.com/course/raspberry-pi-full-stack-raspbian/>

Jose Portilla. (01. 9 2019). Python and Flask Bootcamp: Create Websites using Flask. [Online]

Available At udemy.com:

<https://www.udemy.com/course/python-and-flask-bootcamp-create-websites-using-flask/>

Les, (25, 6 2017), DS18B20 Temperature Sensor With Python Raspberry Pi [Online]

Available At bigl.es:

<https://bigl.es/ds18b20-temperature-sensor-with-python-raspberry-pi/>

Alessandro Maggio,(07, 6 2018), How to create Python Modules, the complete tutorial. [Online]

Available ictshore.com:

<https://www.ictshore.com/python/create-python-modules-tutorial/>

Benne de Bakke,(16, 6 2019), Waterproof JSN-SR04T Ultrasonic Distance Sensor with Arduino Tutorial. [Online]

Available At makerguides.com:

<https://www.makerguides.com/jsn-sr04t-arduino-tutorial/>

**LAMPIRAN**

|  |
| --- |
| **#1.script app.py**  from flask import Flask, render\_template, request, session, redirect, url\_for  from flask\_bootstrap import Bootstrap  from flask\_socketio import SocketIO, send, emit  from AquariumAutomation import DSTemp, JSNDistance, Autowc, CurrentACS, SENTurbidity  from flask\_apscheduler import APScheduler  from threading import Thread, Event  from flask\_wtf import FlaskForm  from wtforms import StringField, PasswordField, BooleanField  from wtforms.validators import InputRequired, Email, Length  from flask\_sqlalchemy import SQLAlchemy  from werkzeug.security import generate\_password\_hash, check\_password\_hash  from flask\_login import LoginManager, UserMixin, login\_user, login\_required, logout\_user, current\_user  import RPi.GPIO as GPIO  import os  import time  import schedule  import glob  app = Flask(\_\_name\_\_)  app.debug = True  app.config['SECRET\_KEY'] = 'fishwebmonitoringsystem'  app.config['SQLALCHEMY\_DATABASE\_URI'] = 'sqlite:////var/www/lab\_app/login.db'  Bootstrap(app)  db = SQLAlchemy(app)  login\_manager = LoginManager()  login\_manager.init\_app(app)  login\_manager.login\_view = 'login'  app.config['DEBUG'] = True  scheduler = APScheduler()  thread = None  #Inisiasi skema table dalam table database  class User(UserMixin, db.Model):  id = db.Column(db.Integer, primary\_key=True)  username = db.Column(db.String(15), unique=True)  email = db.Column(db.String(50), unique=True)  password = db.Column(db.String(80))  #Memuat id user yang sedang login  @login\_manager.user\_loader  def load\_user(user\_id):  return User.query.get(int(user\_id))  #inisiasi atribut dalam login page  class LoginForm(FlaskForm):  username = StringField('username', validators=[InputRequired(), Length(min=6, max=15)])  password = PasswordField('password', validators=[InputRequired(), Length(min=6, max=80)])  remember = BooleanField('remember me')  #Inisiasi mode Bus 1-Wire  os.system('modprobe w1-gpio')  os.system('modprobe w1-therm')  #Tinggi Sensor jika tank Kosong  mainTankDistance = 72 #calibrate this  secondTankDistance = 65 #calibrate this  #cek device sensor suhu  base\_dir = '/sys/bus/w1/devices/'  #sensor temperature Akuarium  device\_folder1 = glob.glob(base\_dir + '28\*')[0]  device\_file1 = device\_folder1 + '/w1\_slave'  #sensor temperature tanki air cadangan  device\_folder2 = glob.glob(base\_dir + '28\*')[1]  device\_file2 = device\_folder2 + '/w1\_slave'  #Set GPIO number mode BCM  GPIO.setmode(GPIO.BCM)  GPIO.setwarnings(False)  #Pin untuk sensor ultrasonic water lvl akuarium  GPIO\_TRIGGER\_MAIN = 15  GPIO\_ECHO\_MAIN = 14  #Pin untuk sensor ultrasonic water lvl tanki air cadangan  GPIO\_TRIGGER\_SECONDARY = 25  GPIO\_ECHO\_SECONDARY = 24  #inisiasi mode pin untuk sensor ultrasonic  GPIO.setup(GPIO\_TRIGGER\_MAIN, GPIO.OUT) # Trigger main  GPIO.setup(GPIO\_TRIGGER\_SECONDARY, GPIO.OUT) # Trigger secondary  GPIO.setup(GPIO\_ECHO\_MAIN, GPIO.IN, pull\_up\_down=GPIO.PUD\_UP) # Echo Main  GPIO.setup(GPIO\_ECHO\_SECONDARY, GPIO.IN, pull\_up\_down=GPIO.PUD\_UP) # Echo Secondary  #Membuat pin stand by di posisi LOW  GPIO.output(GPIO\_TRIGGER\_MAIN, False)  GPIO.output(GPIO\_TRIGGER\_SECONDARY, False)  # inisiasi pin relay  pins = {  17 : {'name' : 'Pompa Utama', 'state' : GPIO.HIGH},  27 : {'name' : 'Pompa Pengisap', 'state' : GPIO.HIGH},  22 : {'name' : 'Pompa Pengisi', 'state' : GPIO.HIGH},  23 : {'name' : 'Lampu', 'state' : GPIO.HIGH}  }  # Mematikan aliran listrik menggunakan relay mode Normally Open  for pin in pins:  GPIO.setup(pin, GPIO.OUT)  GPIO.output(pin, GPIO.HIGH)    @app.route('/login', methods=['GET', 'POST'])  def login():  form = LoginForm()  if form.validate\_on\_submit():  user = User.query.filter\_by(username=form.username.data).first()  if user:  if user.password == form.password.data:  login\_user(user, remember=form.remember.data)  return redirect(url\_for('main'))    return render\_template('login.html', form=form)    @app.route('/logout')  @login\_required  def logout():  logout\_user()  return redirect(url\_for('login'))  @app.route("/home")  @login\_required  def main():  # Menyimpan pin-pin relay dan memasukkan nilai "state"nya ke dalam Dictionary bernama "pins" yang telah diinisiasi sebelumnya  for pin in pins:  pins[pin]['state'] = GPIO.input(pin)    # memasukkan Dictionary "pins" ke dalam dictionary "templateData":  templateData = {  'pins' : pins  }    var\_mainTankWaterLvl = JSNDistance.measureTank(GPIO\_TRIGGER\_MAIN,GPIO\_ECHO\_MAIN,mainTankDistance)  var\_secondaryTankWaterLvl = JSNDistance.measureTank(GPIO\_TRIGGER\_SECONDARY,GPIO\_ECHO\_SECONDARY, secondTankDistance)  var\_mainTankTemperature = DSTemp.read\_temp(device\_file1)  var\_secondaryTankTemperature = DSTemp.read\_temp(device\_file2)    hour, minutes = CurrentACS.get\_battery\_estimated\_life(50,CurrentACS.get\_current\_values(1),20)    return render\_template('home.html',mainTankWaterLvl = var\_mainTankWaterLvl, secondaryTankWaterLvl = var\_secondaryTankWaterLvl ,mainTankTemperature = var\_mainTankTemperature, secondaryTankTemperature = var\_secondaryTankTemperature ,\*\*templateData, hour=str("%.0f" % hour), minutes=str("%.0f" % minutes))  scheduler.add\_job(id = 'waterchange', func = Autowc.detect\_water\_change, trigger = 'interval', hour = 24)  scheduler.start()  if \_\_name\_\_ == "\_\_main\_\_":  app.run(host='0.0.0.0', port=8080,debug=True)  **# 2.script login.html**  {% extends "bootstrap/base.html" %}  {% import "bootstrap/wtf.html" as wtf %}  {% block title %}  Login  {% endblock %}  {% block styles %}  {{super()}}  <link rel="stylesheet" href="{{url\_for('.static/css', filename='signin.css')}}">  {% endblock %}  {% block content %}  <div class="container">  <form class="form-signin" method="POST" action="/login">  <h2 class="form-signin-heading">Please sign in</h2>  {{ form.hidden\_tag() }}  {{ wtf.form\_field(form.username) }}  {{ wtf.form\_field(form.password) }}  {{ wtf.form\_field(form.remember) }}  <button class="btn btn-lg btn-primary btn-block" type="submit">Sign in</button>  </form>  </div> <!-- /container -->  {% endblock %}  **# 3.script home.html**  <!DOCTYPE HTML>  <html>  <head>  <meta charset="utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=yes">    <meta http-equiv="refresh" content="15">  <title>Fish Monitoring System</title>    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">    </head>  <body>  <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js" integrity="sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n" crossorigin="anonymous"></script>  <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></script>  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js" integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6" crossorigin="anonymous"></script>    <h1 id="titleWeb" align="center">Monitoring Fish System</h1>  <a href="/logout" class="btn btn-primary btn-lg btn-block" role="button">LOGOUT</a></div>  <br/>  <br/>  <div class="container">  <div class="card-deck">  <div class="card">  <img src="/static/images/temperature.png" class="card-img-top" alt="...">  <div class="card-body">  <h5 class="card-title">Temperature Akuarium</h5>  <p id="mainTankTemp" class="card-text">Temperatur Air Akuarium : {{mainTankTemperature}} C</p>  </div>  </div>  <div class="card">  <img src="/static/images/temperature.png" class="card-img-top" alt="...">  <div class="card-body">  <h5 class="card-title">Temperatur Air Cadangan</h5>  <p class="card-text">Temperatur Air Cadangan : {{secondaryTankTemperature}} C</p>  </div>  </div>  <div class="card">  <img src="/static/images/waterLvlWhite.png" class="card-img-top" alt="...">  <div class="card-body">  <h5 class="card-title">Tinggi Air Akuarium</h5>  <p class="card-text">Tinggi Air Akuarium : {{mainTankWaterLvl}} cm</p>  </div>  </div>  <div class="card">  <img src="/static/images/waterLvlWhite.png" class="card-img-top" alt="...">  <div class="card-body">  <h5 class="card-title">Tinggi Air Cadangan</h5>  <p class="card-text">Tinggi Air Cadangan : {{secondaryTankWaterLvl}} cm</p>  </div>  </div>  <div class="card">  <img src="/static/images/battery.png" class="card-img-top" alt="...">  <div class="card-body">  <h5 class="card-title">Daya Baterai</h5>  <p id="mainTankTemp" class="card-text">Durasi Operasi Baterai : Sekitar {{hour}} jam {{minutes}} menit</p>  </div>  </div>    </div>  <div class="card-deck">  <div class="card">  <img src="/static/images/waterPump.png" class="card-img-top" alt="...">  <div class="card-body">  <h5 class="card-title">Status Pompa Utama</h5>  <p>{{ pins[17].name }}  {% if pins[17].state == true %}  : <strong>off</strong></p>  {% else %}  : <strong>on</strong></p>  {% endif %}  </div>  </div>  <div class="card">  <img src="/static/images/waterPump.png" class="card-img-top" alt="Status Pompa Pengisap">  <div class="card-body">  <h5 class="card-title">Status Pompa Pengisap</h5>  <p>{{ pins[27].name }}  {% if pins[27].state == true %}  : <strong>off</strong></p>  {% else %}  : <strong>on</strong></p>  {% endif %}  </div>  </div>  <div class="card">  <img src="/static/images/waterPump.png" class="card-img-top" alt="Status Pompa Pengisi">  <div class="card-body">  <h5 class="card-title">Status Pompa Pengisi</h5>  <p>{{ pins[22].name }}  {% if pins[22].state == true %}  : <strong>off</strong></p>  {% else %}  : <strong>on</strong></p>  {% endif %}  </div>  </div>  <div class="card">  <img src="/static/images/lamp.png" class="card-img-top" alt="Status Lampu Aquarium">  <div class="card-body">  <h5 class="card-title">Status Lampu Aquarium</h5>  <p>{{ pins[23].name }}  {% if pins[23].state == true %}  : <strong>off</strong></p>  {% else %}  : <strong>on</strong></p>  {% endif %}  </div>  </div>  <div class="card">  <img src="/static/images/electricity.png" class="card-img-top" alt="...">  <div class="card-body">  <h5 class="card-title">Aliran Listrik Menyala</h5>  <p id="mainTankTemp" class="card-text">Durasi Pengisian Baterai : {{estimatedChargeTimeLeft}}</p>  </div>  </div>  </div>  </div>    </body>    </html>  **# 4. Script DSTemp.py**  def read\_temp\_raw(device\_file):  f = open(device\_file, 'r')  lines = f.readlines()  f.close()  return lines    def read\_temp(device\_file):  lines = read\_temp\_raw(device\_file)  while lines[0].strip()[-3:] != 'YES':  time.sleep(0.2)  lines = read\_temp\_raw(device\_file)  equals\_pos = lines[1].find('t=')  if equals\_pos != -1:  temp\_string = lines[1][equals\_pos+2:]  temp\_c = float(temp\_string) / 1000.0 # Get the tempereture in celcius  temp\_f = temp\_c \* 9.0 / 5.0 + 32.0 # Get the tempereture in fahrenheit    return str("%.2f" % temp\_c)  **# 5. Script SENTurbidity.py**  import time  import board  import busio  import math  import adafruit\_ads1x15.ads1115 as ADC  from adafruit\_ads1x15.analog\_in import AnalogIn  # Create the I2C bus  i2c = busio.I2C(board.SCL, board.SDA)  # Create the ADC object using the I2C bus  adc = ADC.ADS1115(i2c)  dirty\_water\_voltage = 2.85  def check\_turbidity\_values(isWaterChanging):  #channel 0 in ADS1115 module  channel = AnalogIn(adc, ADC.P0)  if(channel.voltage >= dirty\_water\_voltage):  return True  **# 6. Script JSNDistance**  import RPi.GPIO as GPIO  import time  TRIGGER\_TIME = 0.00001  MAX\_TIME = 0.004 # max time waiting for response in case something is missed  #Set GPIO number mode BCM  GPIO.setmode(GPIO.BCM)  # This function measures a distance  def measureTank(triggerPin,echoPin,bottomTankDistance):  # Pulse the trigger/echo line to initiate a measurement  GPIO.output(triggerPin, True)  time.sleep(TRIGGER\_TIME)  GPIO.output(triggerPin, False)  # ensure start time is set in case of very quick return  start = time.time()  timeout = start + MAX\_TIME  # set line to input to check for start of echo response  while GPIO.input(echoPin) == 0 and start <= timeout:  start = time.time()  if(start > timeout):  return "out of range"  stop = time.time()  timeout = stop + MAX\_TIME  # Wait for end of echo response  while GPIO.input(echoPin) == 1 and stop <= timeout:  stop = time.time()  if(stop <= timeout):  elapsed = stop-start  distance = float(elapsed \* 34300)/2.0  else:  return "out of range"  return str("%.2f" % (bottomTankDistance-distance))  **# 7. Script AutoWC**  #script  from AquariumAutomation import JSNDistance  import RPi.GPIO as GPIO  import time  import datetime  from AquariumAutomation import SENTurbidity  #Set GPIO number mode BCM  GPIO.setmode(GPIO.BCM)  GPIO.setwarnings(False)  #HIGH = MENYALA  pompa\_utama = 17  pengisap = 27  pengisi = 22  lampu = 23  GPIO.setup(pompa\_utama, GPIO.OUT) #pompa utama  GPIO.setup(pengisap, GPIO.OUT) #pompa pengisap  GPIO.setup(pengisi, GPIO.OUT) #pompa pengisi  GPIO.setup(lampu, GPIO.OUT) #Lampu  #Pin untuk sensor ultrasonic water lvl akuarium  GPIO\_TRIGGER\_MAIN = 15  GPIO\_ECHO\_MAIN = 14  #Pin untuk sensor ultrasonic water lvl tanki air cadangan  GPIO\_TRIGGER\_SECONDARY = 25  GPIO\_ECHO\_SECONDARY = 24  #inisiasi mode pin untuk sensor ultrasonic  GPIO.setup(GPIO\_TRIGGER\_MAIN, GPIO.OUT) # Trigger main  GPIO.setup(GPIO\_TRIGGER\_SECONDARY, GPIO.OUT) # Trigger secondary  GPIO.setup(GPIO\_ECHO\_MAIN, GPIO.IN, pull\_up\_down=GPIO.PUD\_UP) # Echo Main  GPIO.setup(GPIO\_ECHO\_SECONDARY, GPIO.IN, pull\_up\_down=GPIO.PUD\_UP) # Echo Secondary  GPIO.output(GPIO\_TRIGGER\_MAIN, False)  GPIO.output(GPIO\_TRIGGER\_SECONDARY, False)  #Tinggi Sensor jika tank Kosong  mainTankDistance = 72 #calibrate this  secondTankDistance = 65 #calibrate this    maxWater = JSNDistance.measureTank(GPIO\_TRIGGER\_MAIN,GPIO\_ECHO\_MAIN,mainTankDistance) #mengambil nilai ketinggian  waterVolumetoChange = float(maxWater) / 15 # mengganti 1/15 volume air di dalam tank  volumeAfterDrain = float(maxWater) - float(waterVolumetoChange)  isTankDrained = False  isWaterChangeCompleted = False  day\_scheduled = "Monday"  def detect\_water\_change():  now = datetime.datetime.now()  today = (now.strftime("%A"))    if(today == day\_scheduled):  print("Eksekusi pergantian air metode jadwal")  start\_water\_changing()  elif (chan\_v >= chan\_dirty):  print("Eksekusi pergantian air metode kekeruhan")  elif (chan\_v <= chan\_dirty):  print("Air aman")    def start\_water\_changing():  while isWaterChangeCompleted == False:  time.sleep(2)  if(isTankDrained == False):  drain\_half\_tank()  else :  fill\_full\_tank()  if(isWaterChangeCompleted == True):  print("Water Change completed")  break    def drain\_half\_tank():  # half = maxWater/2 #kode asli  if(float(JSNDistance.measureTank(GPIO\_TRIGGER\_MAIN,GPIO\_ECHO\_MAIN,mainTankDistance)) > float(volumeAfterDrain)):  GPIO.output(pengisap, GPIO.LOW) #Nyalakan pompa pengisap  time.sleep(4)  print("penyedotan sedang berlangsung...")  print("Target volume air :")  print(volumeAfterDrain)  print("Ketinggian Air : ")  print(JSNDistance.measureTank(GPIO\_TRIGGER\_MAIN,GPIO\_ECHO\_MAIN,mainTankDistance))  print("\n\n")  else :  GPIO.output(pengisap, GPIO.HIGH) #Matikan pompa pengisap  globals()['isTankDrained'] = True  print("penyedotan telah selesai!\n\n")  print("\n\n")  time.sleep(1)    def fill\_full\_tank():  if(float(JSNDistance.measureTank(GPIO\_TRIGGER\_MAIN,GPIO\_ECHO\_MAIN,mainTankDistance)) < float(maxWater)):  GPIO.output(pengisi, GPIO.LOW) #Nyalakan pompa pengisi  time.sleep(2)  print("pengisian sedang berlangsung...\n")  print("Ketinggian Air : ")  print(JSNDistance.measureTank(GPIO\_TRIGGER\_MAIN,GPIO\_ECHO\_MAIN,mainTankDistance))  print(" Target volume air :")  print(maxWater)  print("\n\n")  else :  GPIO.output(pengisi, GPIO.HIGH) #Matikan pompa pengisi  print("mematikan pompa pengisi...")  globals()['isWaterChangeCompleted'] = True  **# 8. Script CurrentACS.py**  import time  import board  import busio  import math  import adafruit\_ads1x15.ads1115 as ADC  from adafruit\_ads1x15.analog\_in import AnalogIn  # Create the I2C bus  i2c = busio.I2C(board.SCL, board.SDA)  # Create the ADC object using the I2C bus  adc = ADC.ADS1115(i2c)  #calibrate this according to the sensors. In this case, I use the 5A version  sensitivity = 185 # mv/A for ACS712 5A version, 100 mV/A for 20A version, and 66 mv/A for 30A version  #jika tak ada arus, tegangan harusnya 0  offsetVoltage = 0.5  def get\_current\_values(channel\_num):  chan = ADC.P1  if (channel\_num == 1):  chan = ADC.P1  if(channel\_num == 2):  chan = ADC.P2    channel = AnalogIn(adc,chan)  current = (channel.voltage - offsetVoltage) / sensitivity  return current  def get\_battery\_estimated\_life(battery\_power, current\_flow, battery\_deficiency):  usage\_time = battery\_power/current\_flow  deficiency = usage\_time\*(battery\_deficiency/100)  estimated\_usage\_time = usage\_time - deficiency  minutes = (estimated\_usage\_time % 1) \* 60  hour = estimated\_usage\_time - (estimated\_usage\_time % 1)  #  # print("Hour : ")  # print(str("%.0f" % hour))  # print("Minutes : ")  # print(str("%.0f" % minutes))  return hour, minutes |