

BAB 3. DESAIN SISTEM

3.1 Analisis

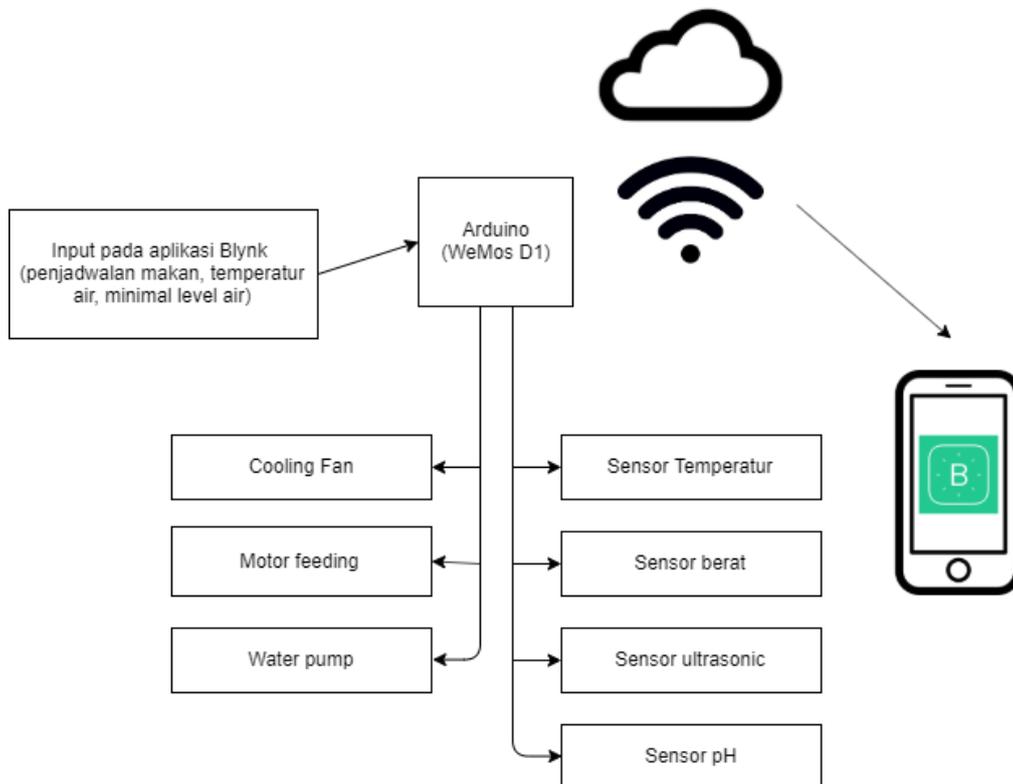
Tujuan utama pembuatan sistem ini adalah untuk menjadi salah satu solusi bagi pemilik ikan terhadap ketidak-tersedianya penitipan ikan. Diharapkan dengan adanya sistem ini dapat memudahkan pemilik ikan dalam mengontrol *aquarium* serta memberi makan ikan dalam jarak jauh. Kondisi yang dapat dikontrol dalam sistem ini terdiri dari 3 jenis yaitu temperatur, ketinggian air, dan pemberian makan. Untuk pemberian makan dilengkapi dengan fitur penjadwalan makan yaitu pengaturan jumlah porsi serta jam pemberian makan.

Untuk membentuk sistem ini, dibutuhkan beberapa perangkat seperti WeMos D1, sensor-sensor (sensor temperatur, sensor *ultrasonic*, sensor pH, serta sensor *load cell*), *relay*, *motor stepper*, *Real Time Clock* (RTC), *fan cooler*, dan *water pump*. WeMos D1 berfungsi sebagai pusat yang mengendalikan perangkat lain. Sensor-sensor berfungsi sebagai pendeteksi perubahan suatu kondisi pada *aquarium*. *Motor stepper* berfungsi sebagai penggerak alat pemberian makan. Sedangkan *Real Time Clock* berfungsi sebagai pendeteksi satuan waktu untuk dapat dikenali oleh WeMos D1. *Board* yang dipilih untuk sistem ini adalah WeMos D1 dengan fitur *WiFi* secara *built-in*. Semua data baik data-data yang dihasilkan oleh sensor maupun data *user* dan data *input* hewan peliharaan akan disimpan pada *database*. *Database* yang digunakan pada sistem ini adalah *database MySQL*.

3.2 Desain

Untuk desain sistem yang digunakan pada skripsi ini adalah desain perangkat *hardware*, desain *software/ User Interface* (UI), dan desain *database*. Gambar di bawah ini merupakan garis besar sistem yang akan dibuat. Sistem ini tersusun dari WeMos sebagai pusat komputasi dan *power supply*, 4 perangkat sensor (sensor temperatur, sensor *load cell*, sensor *ultrasonic*, dan sensor pH), 3 perangkat *aquator* (*cooling fan*, *motor feeding*, dan *water pump*), serta *Real Time*

Clock (RTC) sebagai penunjuk waktu dan *relay module* untuk penghubung atau pemutus aliran listrik yang ditujukan untuk *aquator*.



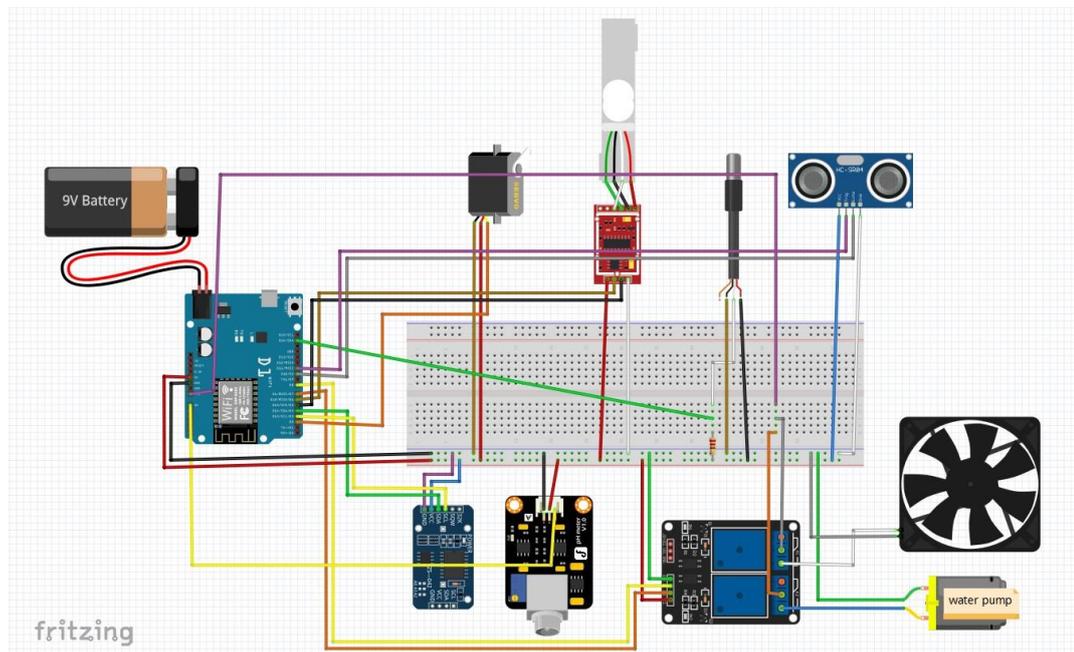
Gambar 3.1 Desain Sistem

Sistem yang ditunjukkan pada gambar 3.1 di atas bertujuan untuk menjalankan dan mengontrol beberapa *aquator* melalui aplikasi *Blynk* pada *Android device* menggunakan koneksi internet. Sistem ini dimulai dengan beberapa *input* yang dimasukkan oleh *user* pada aplikasi *Blynk*. *Input* yang dimasukkan oleh *user* akan dijadikan sebagai kondisi parameter yang akan digunakan sebagai pemicu jalannya *aquator*. Berbagai *aquator* dan sensor dikontrol oleh WeMos sebagai pusat sehingga diperlukan koneksi internet sebagai penghubung WeMos dan aplikasi *Blynk*. Kemudian WeMos akan menyalakan sensor sehingga sensor akan membaca kondisi *aquarium* sehingga menghasilkan *output data* dan hasil *output data* akan disimpan pada *MySQL database*. *Output data* akan dibandingkan dengan *input* dari *user* pada aplikasi *Blynk* dan hasil perbandingan tersebut akan menentukan *aquator* akan dinyalakan atau dimatikan.

3.2.1 Desain *Hardware*

Perangkat yang digunakan dalam sistem ini terdiri dari 10 perangkat yaitu *microcontroller* WeMos D1, 4 sensor, 3 *aquator*, serta *relay module* dan *Real Time Clock* (RTC). Oleh karena itu, dengan banyaknya perangkat yang digunakan pada penelitian ini maka dibutuhkan perencanaan desain *hardware* yang jelas, lengkap, dan terperinci agar tidak menghambat proses dalam pembuatan sistem. Pada penelitian ini, desain *hardware* dibagi menjadi desain rangkaian listrik dan desain *flowchart*. Desain rangkaian listrik digunakan untuk menentukan rangkaian listrik (*wiring*) perangkat sebelum melakukan perangkaian sistem. Sedangkan desain *flowchart* digunakan untuk menentukan alur proses/urutan sistem bekerja dari menyalakan sensor hingga mematikan sensor. Desain *flowchart* yang didesain untuk setiap sensor yang akan digunakan pada penelitian ini.

3.2.1.1 Desain Rangkaian Listrik



Gambar 3.2 Desain Rangkaian Listrik

Gambar 3.2 di atas merupakan desain dari rangkaian listrik dalam sistem yang akan dibuat. Desain rangkaian listrik diperlukan untuk mempermudah proses pembuatan sistem sehingga tidak terjadi kesalahan dalam merangkai sistem.

Perangkat-perangkat yang digunakan dalam sistem ini terdiri dari 10 perangkat yaitu *microcontroller*, 4 sensor, 3 *aquator*, serta *relay module* dan *Real Time Clock* (RTC). Perangkat *microcontroller* yang digunakan dalam penelitian ini adalah WeMos D1. Sensor-sensor yang digunakan dalam penelitian ini adalah sensor temperatur, sensor *ultrasonic*, sensor *load cell*, dan sensor pH sedangkan *aquator* yang digunakan adalah *cooling fan*, *motor feeding*, dan *water pump*.

Pin yang digunakan pada *Real Time Clock* adalah SCL, SDA, VCC, dan GND. Baik pin SCL dan SDA masing-masing digunakan sebagai *serial clock* dan *serial data* untuk *I2C interface*. Pin SCL dan SDA masing-masing dihubungkan dengan pin *digital* (D3 dan D4) yang ada di WeMos. Untuk pin VCC dihubungkan dengan pin 3,3V WeMos, dan untuk pin GND dihubungkan dengan pin GND WeMos.

Pada *relay module* terdapat banyak pin pada kedua sisi (Kori et al., 2019). Pada salah satu sisi *relay module* terdapat pin *ground* (GND), *Input 1* (IN1), *Input 2* (IN2), dan VCC. Sisi kanan pada *relay module* ini membutuhkan tegangan kecil sehingga dapat dihubungkan dengan perangkat *microcontroller*. Pin IN1 dan pin IN2 masing-masing dihubungkan dengan pin *digital* (D8 dan D7). Untuk pin GND dihubungkan dengan pin GND WeMos, dan untuk pin VCC dihubungkan dengan pin 5V WeMos. Pada sisi lain *relay module*, pin yang digunakan hanya pin *Normally Open* (NO) dan *Common* (COMM). Pin NO (yang terdapat pada bagian kiri) dihubungkan dengan pin VCC atau pin (+) pada perangkat elektronik sedangkan pin COMM (yang terdapat pada bagian tengah) dihubungkan dengan pin Vin pada WeMos. Untuk pin GND atau pin (-) pada perangkat elektronik dapat dihubungkan dengan pin GND pada WeMos.

DS18B20 *waterproof temperature sensor* memiliki 3 pin yaitu pin GND/*Ground*, DQ, dan VDD. Pin GND berfungsi sebagai *Ground*, pin DQ sebagai *1-Wire Data Bus* yang menggunakan *1-Wire method* sehingga akan menghasilkan *output data* nilai temperatur, dan pin VDD sebagai menyediakan *power*. Pin GND pada DS18B20 dihubungkan dengan pin GND pada WeMos. Pin DQ pada DS18B20 dihubungkan dengan pin *digital* pada WeMos. Sedangkan pin VDD pada DS18B20 dihubungkan dengan pin 5V pada WeMos.

Ultrasonic sensor memiliki 4 pin yaitu pin VCC, pin Trig/ Trigger (input), pin Echo (output), dan pin GND/ Ground. Pin VCC digunakan sebagai *power supply* pada *ultrasonic sensor* yang dihubungkan dengan pin 5 Volts (5V) pada WeMos. Pin Trig digunakan sebagai *trigger ultrasonic sound pulse*. Sedangkan pin Echo digunakan untuk memancarkan *pulse* ketika *signal* yang direfleksikan diterima (panjang dari *pulse* sebanding dengan waktu yang dibutuhkan agar sinyal yang ditransmisikan terdeteksi). Pin Trig dan pin Echo dihubungkan dengan pin I/O *digital* pada WeMos. Pin Ground (GND) digunakan untuk dihubungkan dengan pin *Ground* (GND) pada WeMos.

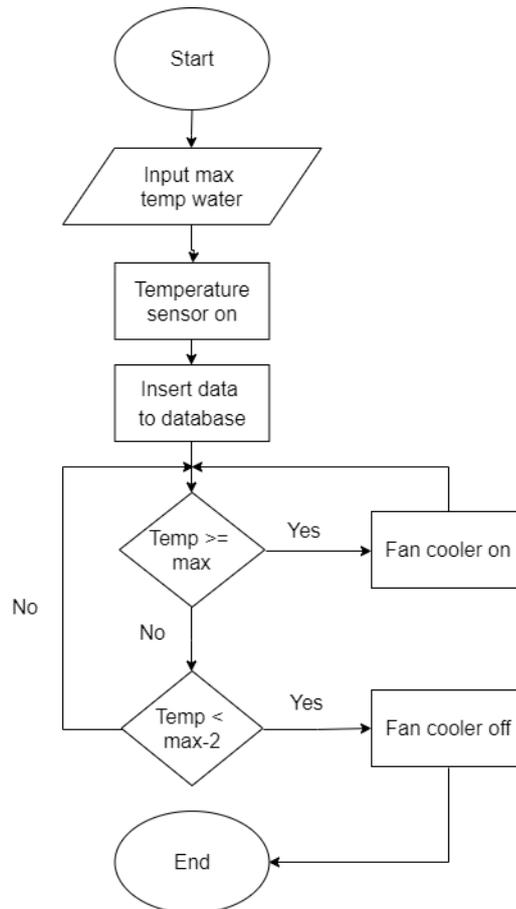
Pada *Load Cell* terdapat kabel berwarna merah, hitam, putih, dan hijau. Kabel merah masuk ke pin *Excitation+* (E+) yang berarti aliran listrik positif atau VCC. Kabel hitam masuk ke pin *Excitation-* (E-) yang berarti aliran listrik negatif atau GND. Kabel putih masuk ke pin *Amplifier-* (A-) dan kabel hijau masuk ke pin *Amplifier+* (A+) yang menandakan akan memakai pin *analog* pada Arduino sebagai *input*. Untuk terhubung ke Arduino, maka pin sebelah kanan dari HX711 harus dihubungkan terlebih dahulu ke Arduino. Pin VCC dihubungkan dengan pin 5V Arduino. Pin *Ground* (GND) dihubungkan dengan pin *Ground* (GND) di Arduino. Pin TX dan pin RX dihubungkan dengan pin *digital* di Arduino.

Sedangkan untuk sensor pH, 3 pin yang terdapat pada pH sensor yaitu pin GND, pin VCC, dan pin A0/ *Analog*. Pin GND berfungsi sebagai *Ground*, pin VCC sebagai *power supply* pada sensor pH, dan pin *analog* (A) sebagai penyalur data yang didapat dari sensor pH dan menghasilkan *output* berupa sinyal *analog*. Pin *Ground* (GND) pada sensor pH dihubungkan dengan pin *Ground* (GND) pada WeMos. Pin VCC pada sensor pH dihubungkan dengan pin VCC pada WeMos. Sedangkan pin A0 pada sensor pH dihubungkan dengan pin A0 pada WeMos.

3.2.1.2 Desain Flowchart

Desain *Flowchart* dibutuhkan untuk mempermudah pemahaman urutan kerja sistem yang dijelaskan secara *detail* berdasarkan jenis sensor yang akan digunakan dalam sistem. *Flowchart* pada sistem ini terdiri dari *flowchart* sensor temperatur, sensor *ultrasonic*, sensor *load cell*, dan sensor pH.

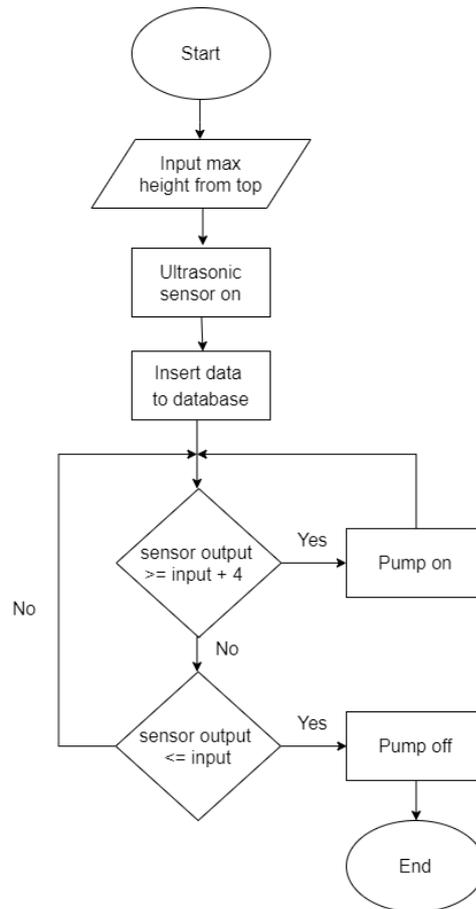
3.2.1.2.1 Sensor Temperatur



Gambar 3.3 *Flowchart* Sensor Temperatur

Gambar 3.3 di atas menunjukkan gambar proses garis besar sensor temperatur bekerja. Sistem ini akan dilengkapi *aquator fan cooler* sebagai pengontrol suhu *aquarium*. *User* akan memasukkan data maksimum dan minimum suhu air. Kemudian sensor temperatur akan dinyalakan dan melakukan pengecekan antara suhu yang di-*input user* dan suhu hasil *output* sensor. *Fan cooler* akan dinyalakan atau dimatikan berdasarkan hasil pengecekan tersebut. Bila suhu hasil *output* sensor lebih besar sama dengan dari suhu yang diinput *user*, maka *fan cooler* dinyalakan. Sebaliknya, bila suhu hasil *output* sensor lebih kecil dari 2°C dari suhu yang diinput *user*, maka *fan cooler* dimatikan.

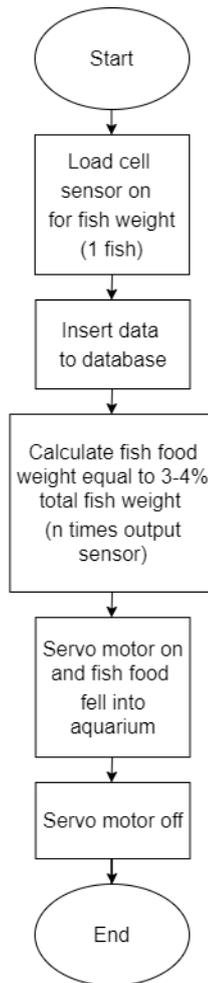
3.2.1.2.2 Sensor *Ultrasonic*



Gambar 3.4 *Flowchart* Sensor *Ultrasonic*

Gambar 3.4 di atas merupakan gambar proses garis besar sensor *ultrasonic* bekerja. Sensor ini dilengkapi dengan *aquator* pompa air sebagai pengontrol ketinggian air pada *aquarium*. *User* akan diminta untuk mengisi batas maksimal jarak antara air *aquarium* dengan tinggi maksimal *aquarium*. Sensor akan diletakkan pada atas *aquarium* sehingga dapat mendeteksi ketinggian air *aquarium*. Jika jarak antara air *aquarium* dengan tinggi maksimal *aquarium* kurang dari sama dengan *input-an user*, maka pompa air akan dimatikan. Sebaliknya jika jarak antara air *aquarium* dengan tinggi maksimal *aquarium* melebihi dari *input-an user* ditambah dengan 4-5 cm, maka pompa air akan dinyalakan sehingga dengan begitu air *aquarium* akan bertambah.

3.2.1.2.3 Sensor *Load Cell*

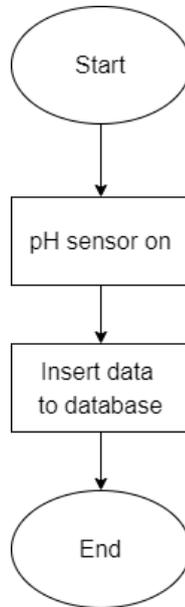


Gambar 3.5 *Flowchart Sensor Load Cell*

Gambar 3.5 di atas merupakan gambar proses garis besar sensor *Load Cell*/ sensor berat bekerja. Langkah pertama yaitu dengan menyalakan sensor *Load Cell* kemudian sensor tersebut akan mengukur berat salah satu ikan. Data tersebut akan dikalkulasi dengan dikalikan dengan jumlah ikan yang berada pada *aquarium*. Kemudian 3-4% dari hasil total berat ikan akan dijadikan sebagai berat makanan ikan. Berat makanan ikan yang diberikan tidak secara langsung ditimbang, melainkan dengan melakukan percobaan untuk mendapatkan seberapa banyak makanan yang jatuh ketika penutup makanan ikan digeser dengan kecepatan tertentu. Percobaan akan dilakukan sebanyak 10 kali untuk setiap kecepatan yang dilakukan pada penutup makanan ikan. Berat makanan ikan yang jatuh akan dicatat dan dikalkulasi sehingga untuk setiap kecepatan pada penutup makanan ikan akan mendapatkan rata-rata berat makanan ikan. Jika didapatkan

range berat ikan yang didapat dari sensor *Load Cell*, maka akan dilakukan kecepatan yang sesuai dari percobaan yang telah dilakukan.

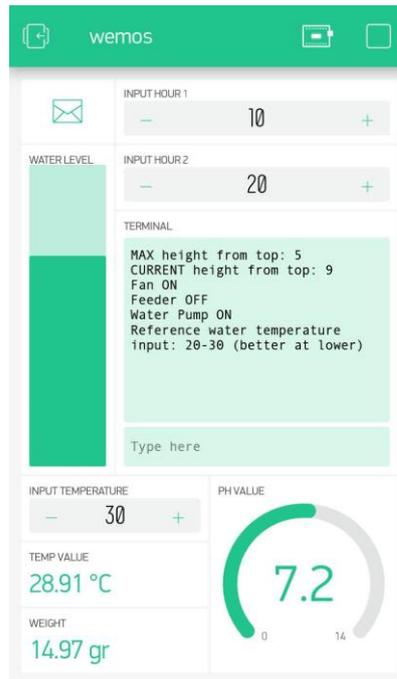
3.2.1.2.4 Sensor pH



Gambar 3.6 *Flowchart* Sensor pH

Gambar 3.6 di atas merupakan gambar proses garis besar sensor pH bekerja. Sensor pH digunakan untuk mengukur pH air pada *aquarium*. Sensor akan dinyalakan secara terus-menerus untuk memantau perubahan pH air *aquarium*. Kemudian hasil data yang didapat dari sensor pH akan dikirim ke *database* melalui koneksi internet. Data yang terdapat pada *database*, akan ditampilkan pada aplikasi *Blynk* dan pada *website*.

3.2.2 Desain Software



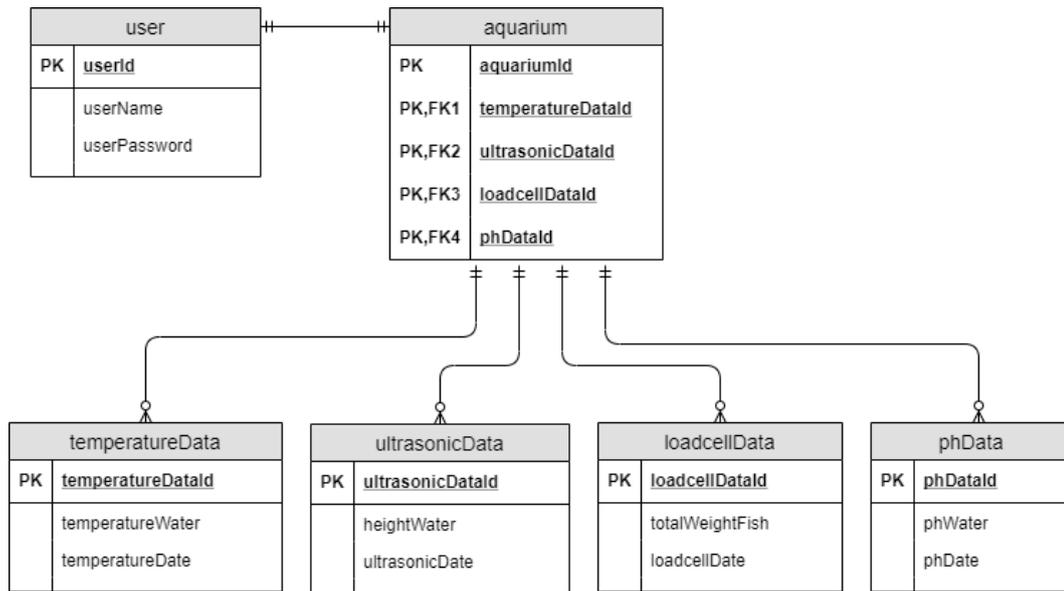
Gambar 3.7 Desain *User Interface* (UI) Aplikasi *Blynk*

Aplikasi yang digunakan untuk mempermudah *user* dalam *monitoring* dan mengontrol sistem ini adalah aplikasi *Blynk*. Aplikasi *Blynk* dapat terhubung dengan beberapa perangkat *microcontroller*, salah satunya yaitu WeMos. Pada aplikasi *Blynk* terdapat beberapa macam jenis *widget* yang tersedia yaitu *widget controllers*, *widget displays*, *widget notifications*, *widget device management*, *widget other*, *widget interface*, dan *widget smartphone sensors*. Untuk sistem ini *widget display*, *widget interface*, dan *widget notification* yang akan dipakai. Untuk desain *User Interface* (UI) pada aplikasi *Blynk* dapat dilihat pada gambar 3.7 di atas.

3.2.3 Desain Database

Database yang digunakan untuk menyimpan data-data yang didapatkan dari sensor pada sistem ini adalah MySQL *database*. Dengan menentukan desain *database* terlebih dulu, maka pembuatan *database* akan lebih efektif karena sudah mengetahui data-data yang akan disimpan. Desain *database* dibagi menjadi *Entity Relationship Diagram* (ERD), *Use Case*, dan *Activity Diagram*.

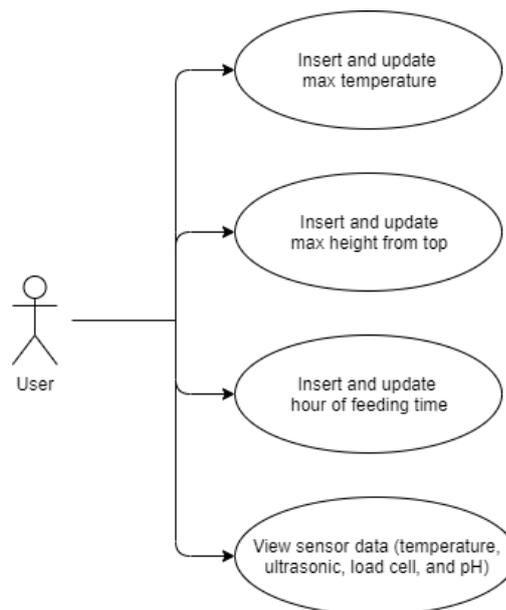
3.2.3.1 Entity Relationship Diagram (ERD)



Gambar 3.8 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) bertujuan untuk merencanakan struktur database berupa entity beserta relasi antar entity. Untuk gambar Entity Relationship Diagram (ERD) dapat dilihat pada gambar 3.8 di atas. Dengan adanya ERD, struktur database sistem dapat disusun dengan matang sebelum mengimplementasikan dalam pembuatan database.

3.2.3.2 Use Case



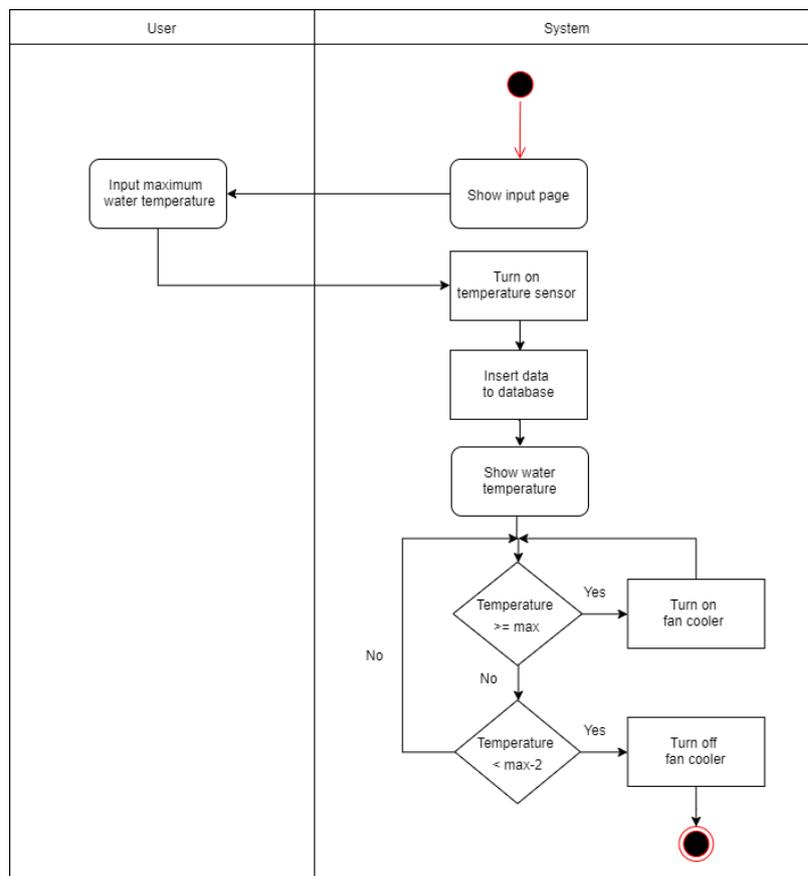
Gambar 3.9 Use Case

Use case tidak disarankan untuk mencatat sistem sesuai dengan urutan kerja sistem, melainkan *use case* bertujuan untuk mencatat *role-role* yang dimiliki oleh aktor. Dalam sistem ini, aktor mengacu pada *user* yang menggunakan aplikasi *Blynk* yang akan mengontrol kondisi *aquarium*. Kondisi *aquarium* yang dikontrol berupa maksimum temperatur, maksimum ketinggian antara air dengan tinggi maksimum *aquarium*, serta jam pemberian makanan ikan. Gambar *use case* dapat dilihat pada gambar 3.9 di atas.

3.2.3.3 Activity Diagram

Berbeda dengan *use case*, *activity diagram* dibuat dengan tujuan untuk menjelaskan diagram berdasarkan runtutan aktifitas yang terjadi pada sistem. *Activity Diagram* dibuat berdasarkan *use case* sistem yang mendeskripsikan *role* dari aktor.

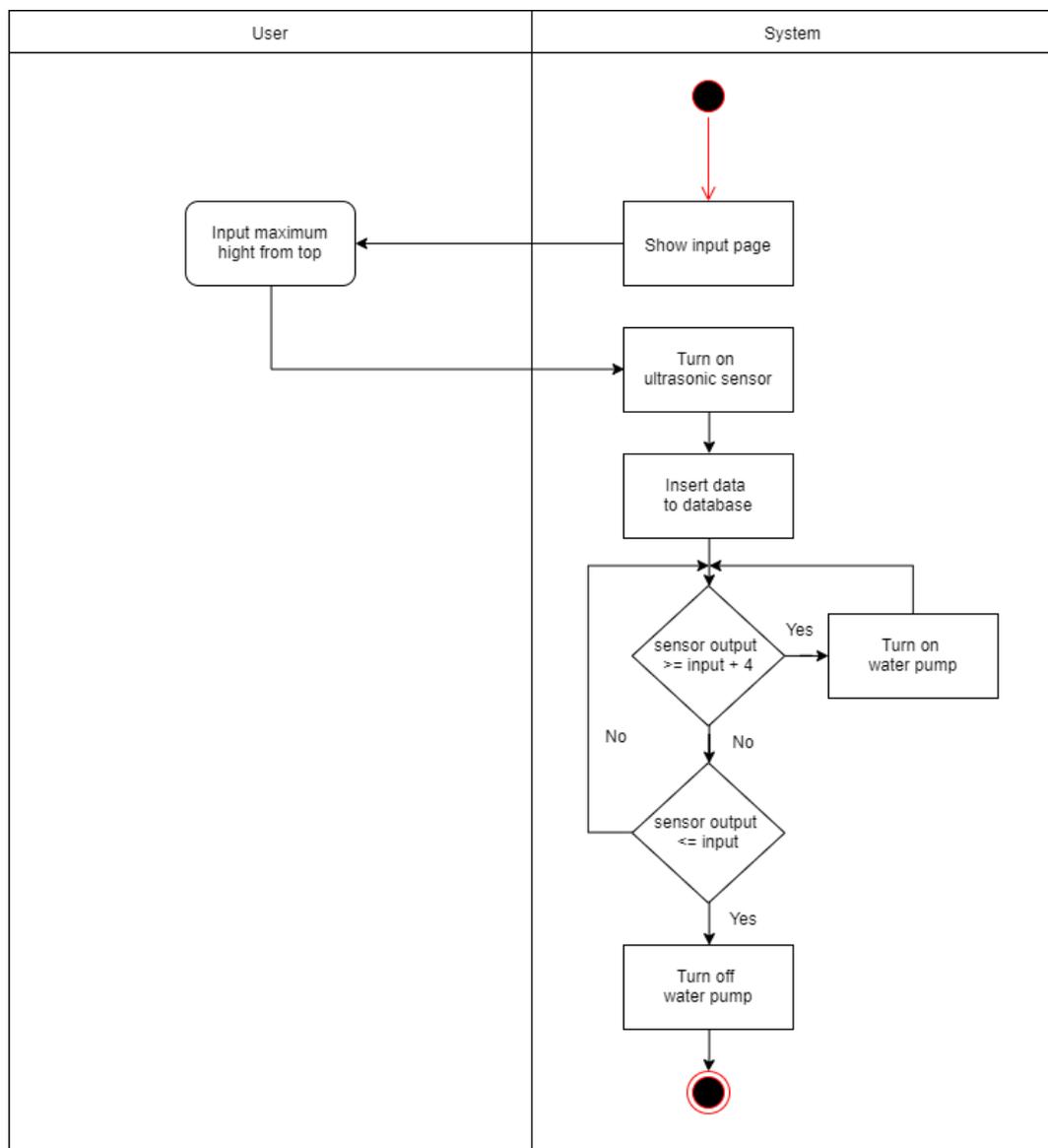
3.2.3.3.1 Sensor Temperatur



Gambar 3.10 Activity Diagram Sensor Temperatur

Proses sensor temperatur dimulai dengan sistem yang menampilkan halaman *input* kemudian *user* akan mengisi *input* yang diperlukan untuk sensor temperatur. Setelah itu sistem akan menyalakan sensor temperatur dan membandingkan data hasil sensor dengan *input*-an dari *user*. Hasil perbandingan antara kondisi tersebut akan menentukan *aquator* (*fan cooling*) akan dinyalakan/dimatikan. *Activity diagram* sensor temperatur dapat dilihat pada gambar 3.10 di atas.

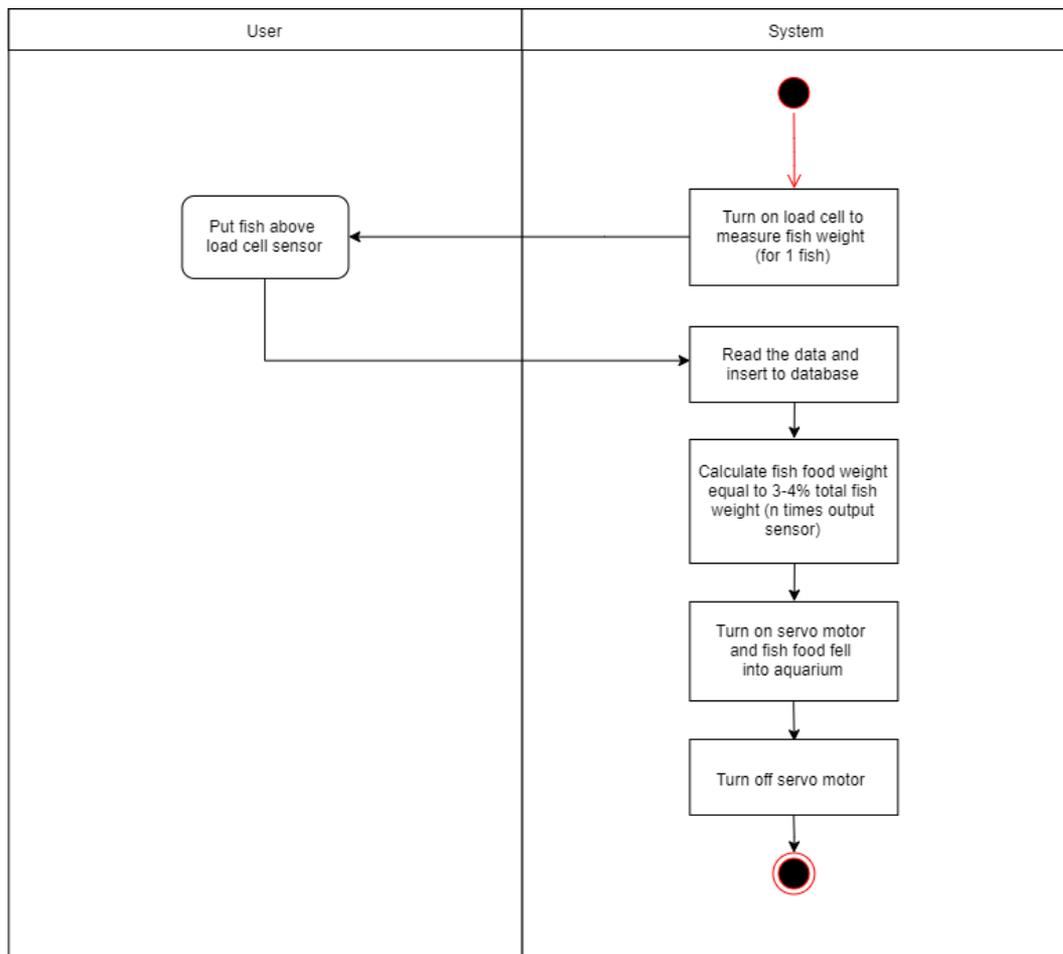
3.2.3.3.2 Sensor Ultrasonic



Gambar 3.11 Activity Diagram Sensor Ultrasonic

Langkah awal sistem sensor *ultrasonic* dimulai dengan sistem yang menampilkan halaman *input* dan *user* memasukkan *input* maksimum tinggi antara air dengan tinggi maksimum *aquarium*. Ketika sensor *ultrasonic* dinyalakan, maka hasil data sensor akan dibandingkan dengan data *input* dari *user*. Hal tersebut akan memicu *aquator* (pompa air) agar dinyalakan atau dimatikan. *Activity diagram* sensor *ultrasonic* dapat dilihat pada gambar 3.11 di atas.

3.2.3.3.3 Sensor Load Cell

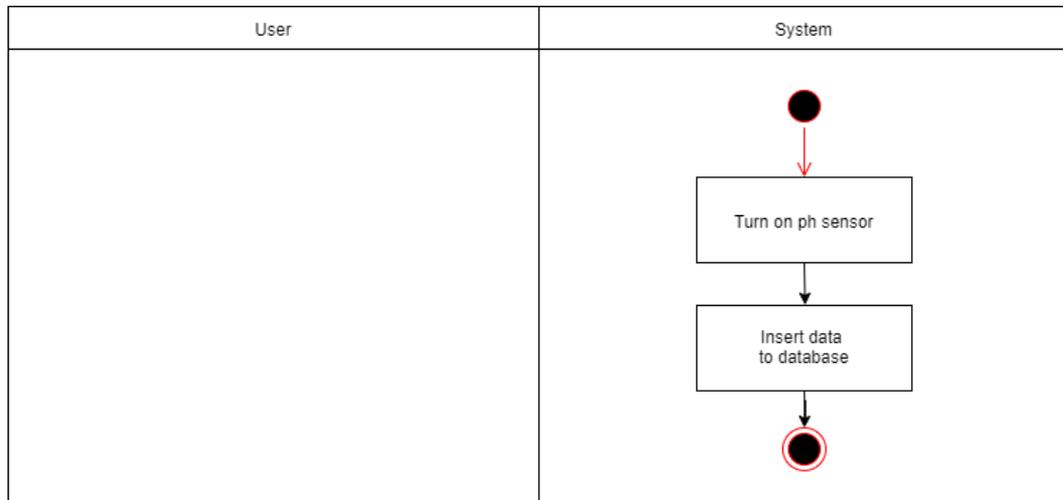


Gambar 3.12 Activity Diagram Sensor Load Cell

Sistem sensor berat/ *load cell* akan dimulai dengan menyalakan sensor *load cell* untuk mengukur berat total ikan. Kemudian *user* akan mengukur total berat ikan dan dilanjutkan dengan menyalakan *servo motor* untuk menuang makanan ikan yang sedang ditimbang serta menyalakan sensor untuk mengukur berat total makanan ikan. Sensor akan menghentikan *servo motor* jika total berat

makanan ikan mencapai 3% dari total berat ikan. *Activity diagram sensor load cell* dapat dilihat pada gambar 3.12 di atas.

3.2.3.3.4 Sensor pH



Gambar 3.13 *Activity Diagram* Sensor pH

Pada awalnya, sistem akan menyalakan sensor pH untuk mengukur pH air pada *aquarium*. Kemudian data yang didapat akan dimasukkan ke dalam tabel untuk sensor pH dalam *database*. Proses tersebut akan terulang setiap 15 menit sekali. Namun sebelum memulai proses pada *activity diagram* 3.13 di atas, sensor pH harus dikalibrasi terlebih dahulu untuk mempertahankan keakuratan pembacaan data.