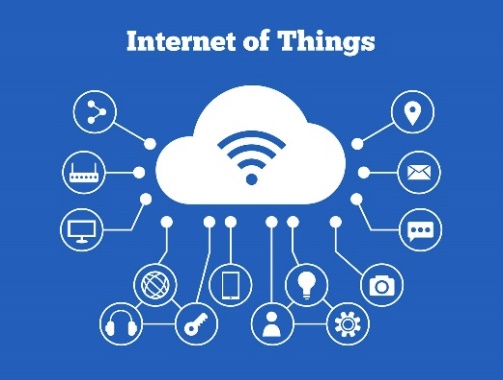
2. Tinjauan Pustaka

2.1. Internet Of Things

Internet of Things (IoT) merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus-menerus. Adapun kemampuan seperti berbagi data, remote control, dan sebagainya, termasuk juga pada benda di dunia nyata. Contohnya bahan pangan, elektronik, koleksi, peralatan apa saja, termasuk benda hidup yang semuanya tersambung ke jaringan lokal dan global melalui sensor yang tertanam dan selalu aktif (Junaidi, 2015).

Menurut Kurniawan et al. (2020), IoT adalah suatu konsep dimana objek tertentu memiliki kemampuan untuk mentransfer data melalui jaringan tanpa memerlukan interaksi dari manusia ke manusia ataupun dari manusia ke perangkat komputer. IoT telah berkembang dari konvergensi teknologi nirkabel, micro-electromechanical systems (MEMS), dan Internet.



**Gambar 2.1 Internet of Things**

(<http://www.djaxtech.com/>)

2.2. Blynk

Blynk adalah platform berbasis Internet of Things (IoT) yang memungkinkan pengguna untuk mengontrol perangkat keras dari jarak jauh melalui aplikasi smartphone. Blynk menyediakan antarmuka yang user-friendly untuk menghubungkan berbagai sensor dan aktuator ke internet, memantau data secara real-time, dan mengontrol perangkat melalui dashboard yang dapat dikustomisasi. (Hidayat, A., & Setiawan, B., 2020).



**Gambar 2.2 Blynk IoT**

(<https://www.getapp.com/>)

2.3. Ikan Mas Koki

Ikan mas koki (Carassius auratus) adalah salah satu jenis ikan hias air tawar yang populer di Indonesia. Ikan ini termasuk dalam famili Cyprinidae dan berasal dari China (Kusrini et al., 2015). Mas koki memiliki berbagai varietas dengan bentuk tubuh dan warna yang beragam, menjadikannya salah satu pilihan favorit di kalangan penggemar ikan hias.



**Gambar 2.3 Ikan Mas Koki**

(<https://buzzingbeeboss.blogspot.com/>)

Menurut Nasichah et al. (2016), ikan mas koki memiliki karakteristik morfologi yang unik, termasuk tubuh yang bulat dan sirip ekor ganda. Warna ikan mas koki bervariasi, mulai dari emas, merah, hitam, hingga kombinasi warna-warna tersebut. Keunikan ini menjadi daya tarik utama bagi para penghobi dan pembudidaya ikan hias.

2.3.1. Habitat Ikan Mas Koki

Ikan mas koki (Carassius auratus) awalnya berasal dari perairan tawar di China dan Jepang. Menurut Kusrini dan Cindelaras (2018), habitat alami ikan mas koki meliputi:

1. Sungai berarus lambat
2. Danau dangkal
3. Kolam alami
4. Rawa-rawa

Di habitat alaminya, ikan mas koki hidup di perairan dengan karakteristik sebagai berikut :

* Suhu air berkisar antara 10-25°C
* pH air antara 6,0-8,0
* Kedalaman air 0,5-3 meter
* Air yang kaya akan vegetasi akuatik

Namun, sebagai ikan hias yang telah mengalami domestikasi selama berabad-abad, ikan mas koki telah beradaptasi dengan berbagai lingkungan buatan seperti akuarium dan kolam hias (Setijaningsih et al., 2017).

2.3.2. Jenis-jenis Ikan Mas Koki

"Ikan mas koki (Carassius auratus) memiliki berbagai jenis yang dibedakan berdasarkan bentuk tubuh, warna, dan sirip. Beberapa jenis yang populer termasuk oranda, ryukin, telescope, dan fantail. Habitat ikan mas koki secara alami adalah perairan tawar dengan aliran air yang tenang, namun sekarang lebih umum ditemukan di akuarium dan kolam hias" (Suryanto, T., & Hartini, Y., 2020).

2.3.2.1. Ikan Mas Koki Oranda

Oranda dikenal dengan "topi" atau wen-nya, yaitu pertumbuhan jaringan berlebih di kepala. Menurut Kusrini et al. (2016), wen ini mulai berkembang saat ikan berusia 3-4 bulan dan terus tumbuh hingga ikan berusia 2 tahun. Oranda memiliki tubuh bulat dengan sirip ekor ganda yang panjang dan mengembang.



**Gambar 2.4 Ikan Mas Koki Oranda**

(<https://4.bp.blogspot.com/>)

2.3.2.2. Ikan Mas Koki Ranchu

Ranchu, juga dikenal sebagai "Raja Goldfish", memiliki tubuh bulat tanpa sirip punggung. Rahardjo et al. (2019) menyatakan bahwa Ranchu memiliki kepala besar dengan wen yang berkembang baik, dan ekor pendek ganda yang menyerupai pompon. Jenis ini memerlukan perawatan khusus karena rentan terhadap masalah pencernaan.



**Gambar 2.5 Ikan Mas Koki Ranchu**

(<https://lifestyle.haluan.co/>)

2.3.2.3. Ikan Mas Koki Ryukin

Ryukin memiliki badan pendek dan bulat dengan punggung yang tinggi dan cembung. Ciri khasnya adalah sirip punggung yang tinggi dan sirip ekor ganda yang panjang. Menurut Setijaningsih dan Umar (2018), Ryukin adalah salah satu jenis yang paling tahan terhadap perubahan suhu.



**Gambar 2.6. Ikan Mas Koki Ryukin**

(<https://www.ruparupa.com/>)

2.3.2.4. Ikan Mas Koki Tosakin

Tosakin memiliki ekor unik berbentuk kipas horizontal yang terbuka. Purnama et al. (2020) menjelaskan bahwa bentuk ekor ini adalah hasil dari mutasi genetik dan seleksi selektif. Tosakin memerlukan kolam atau akuarium yang luas karena bentuk ekornya yang khusus.



**Gambar 2.7 Ikan Mas Koki Tosakin**

(<https://www.ruparupa.com/>)

2.3.2.5. Ikan Mas Koki Teleskop

Teleskop atau Demekin memiliki mata yang menonjol keluar seperti teleskop. Wijaya et al. (2017) menyatakan bahwa mata yang menonjol ini membuat ikan jenis ini rentan terhadap cedera dan infeksi, sehingga memerlukan perawatan ekstra.



**Gambar 2.8 Ikan Mas Koki Teleskop**

**(**<https://www.ruparupa.com/>)

2.3.2.6. Ikan Mas Koki Bubble Eye

Bubble Eye memiliki kantung berisi cairan di bawah mata yang menyerupai gelembung. Menurut Rohmana et al. (2021), kantung ini sangat sensitif dan mudah pecah, sehingga ikan jenis ini harus dipelihara dalam lingkungan yang bebas dari benda tajam.



**Gambar 2.9 Ikan Mas Koki Bubble Eye**

**(**<https://www.ruparupa.com/>)

2.3.2.7. Ikan Mas Koki Lionhead

Lionhead memiliki pertumbuhan daging di kepala yang menyerupai surai singa. Supriyono et al. (2019) menjelaskan bahwa pertumbuhan ini dapat menutupi mata dan mulut ikan, sehingga perlu diperhatikan agar tidak mengganggu penglihatan dan makan ikan.



**Gambar 2.10 Ikan Mas Koki Lionhead**

**(**<https://www.ruparupa.com/>)

2.3.2.8. Ikan Mas Koki Wakin

Wakin adalah jenis yang paling mirip dengan ikan mas liar. Memiliki tubuh ramping dan sirip ekor ganda. Nasichah et al. (2018) menyatakan bahwa Wakin adalah jenis yang paling tahan terhadap penyakit dan perubahan lingkungan.



**Gambar 2.11 Ikan Mas Koki Wakin**

**(**<https://www.ruparupa.com/>)

2.3.2.9. Ikan Mas Koki Pearlscale

Pearlscale memiliki sisik yang menonjol berbentuk mutiara. Menurut Junaidi dan Akmal (2020), bentuk sisik ini membuat ikan jenis ini rentan terhadap infeksi sisik, sehingga memerlukan perawatan air yang sangat baik.



**Gambar 2.12 Ikan Mas Koki Pearlscale**

(<https://www.ruparupa.com/>)

2.4. Ikan Mas Koki Ryukin

2.4.1. Karakteristik Morfologi

**A. Bentuk Tubuh**

Ryukin memiliki badan pendek dan bulat dengan punggung yang tinggi dan cembung. Menurut Kusrini dan Cindelaras (2019), rasio tinggi badan terhadap panjang badan Ryukin bisa mencapai 1:1, memberikan kesan "hunchback" atau bungkuk.



**Gambar 2.13 Ikan Mas Koki Ryukin**

(<https://www.pinterest.com/>)

**B. Sirip**

* Sirip punggung tinggi dan tegak, biasanya dimulai tepat di belakang kepala.
* Sirip ekor ganda (bifurcated) dan panjang, sering mencapai panjang yang sama dengan tubuhnya.
* Sirip anal ganda, menambah keindahan tampilannya.

**C. Kepala**

Ryukin memiliki kepala yang relatif kecil dibandingkan dengan ukuran badannya yang bulat. Beberapa varietas Ryukin memiliki "wen" atau pertumbuhan seperti topi di kepala, meskipun tidak sebesar Oranda (Rahardjo et al., 2020).

**D. Warna**

Ryukin hadir dalam berbagai variasi warna, termasuk merah, putih, hitam, kuning, calico (campuran), dan kombinasi warna-warna tersebut.

2.4.2. Habitat dan Perawatan

Ryukin dapat dipelihara di akuarium atau kolam. Wijaya dan Purnomo (2018) menyarankan kondisi optimal berikut:

* Suhu air: 18-22°C
* pH: 6,8-7,4
* Hardness: 5-19 dGH
* Volume air minimal: 75-100 liter per ikan dewasa

2.4.3. Pakan dan Nutrisi

Ryukin adalah pemakan omnivora. Purnama et al. (2021) merekomendasikan diet yang terdiri dari:

* Pelet komersial khusus ikan mas koki
* Sayuran hijau seperti bayam atau selada
* Makanan hidup seperti cacing darah atau Daphnia (secara terbatas)

2.4.4. Perilaku dan Sosial

Ryukin cenderung berenang di bagian tengah akuarium. Mereka umumnya damai dan dapat dipelihara bersama jenis ikan mas koki lainnya. Namun, karena bentuk tubuhnya yang bulat, Ryukin bisa menjadi perenang yang lambat dibandingkan dengan jenis lain (Supriyono et al., 2020).

2.4.5. Kecerdasan dan Pelatihan

Kusrini dan Maharani (2018) menunjukkan bahwa Ryukin memiliki tingkat kecerdasan yang cukup tinggi untuk ikan hias:

* Dapat mengenali pemiliknya.
* Bisa dilatih untuk makan dari tangan.
* Mampu membedakan warna dan bentuk objek sederhana.

2.5. Mikrokontroller

Mikrokontroler adalah sebuah sistem komputer fungsional dalam sebuah chip. Di dalamnya terkandung sebuah inti prosesor, memori (sejumlah kecil RAM, memori program, atau keduanya), dan perlengkapan input output. Dengan kata lain, mikrokontroler adalah suatu alat elektronika digital yang mempunyai masukan dan keluaran serta kendali dengan program yang bisa ditulis dan dihapus dengan cara khusus (Andrianto & Darmawan, 2020).

Mikrokontroler digunakan dalam produk dan alat yang dikendalikan secara otomatis, seperti sistem kontrol mesin, remote controls, mesin kantor, peralatan rumah tangga, alat berat, dan mainan. Dengan mengurangi ukuran, biaya, dan konsumsi tenaga dibandingkan dengan mendesain menggunakan mikroprosesor memori, dan alat input output yang terpisah, kehadiran mikrokontroler membuat kontrol elektrik untuk berbagai proses menjadi lebih ekonomis (Putra et al., 2021).

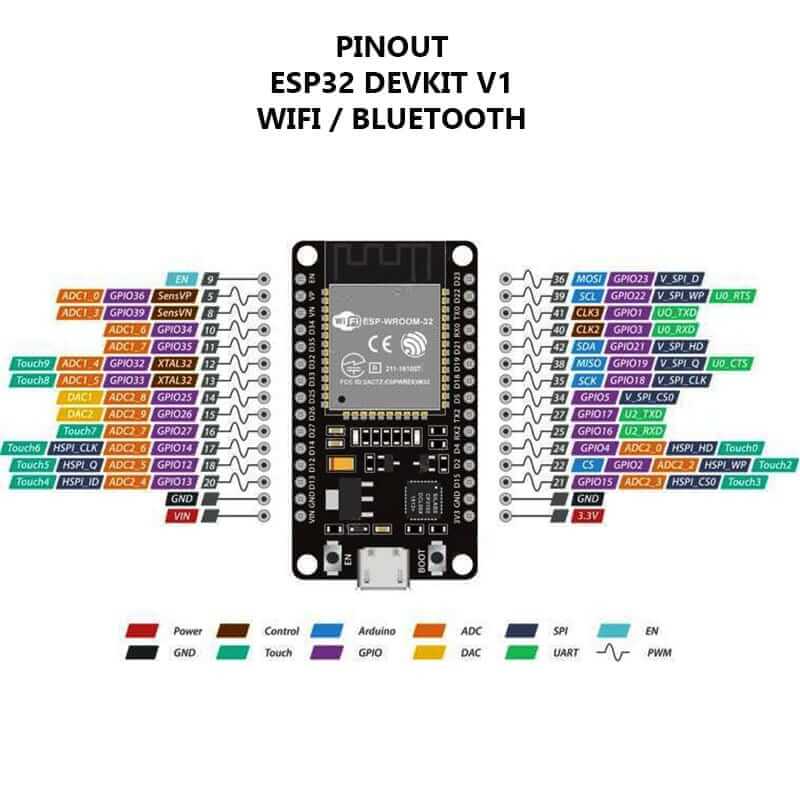
2.5.1. NodeMCU ESP32

ESP32 adalah mikrokontroler yang dirancang oleh Espressif Systems, perusahaan asal Shanghai, Tiongkok. ESP32 merupakan penerus dari ESP8266 dan menawarkan kemampuan yang lebih canggih. Mikrokontroler ini telah menjadi populer di kalangan pengembang IoT (Internet of Things) karena fitur-fiturnya yang kaya dan harganya yang terjangkau (Kurniawan et al., 2021).

Salah satu keunggulan ESP32 adalah dukungan komunitas yang luas dan tersedianya berbagai framework pengembangan, termasuk Arduino IDE, ESP-IDF (Espressif IoT Development Framework), dan MicroPython. Hal ini memudahkan pengembang untuk memulai proyek dengan ESP32, baik untuk pemula maupun profesional (Susanto et al., 2023).

Spesifikasi pin-out pada ESP32 :

1. **GPIO** : 34 pin GPIO multifungsi
2. **ADC** : 18 saluran ADC dengan resolusi hingga 12-bit
3. **DAC** : 2 saluran DAC
4. **Touch Sensor** : 10 touch sensor kapasitif
5. **PWM** : 16 saluran PWM
6. **I2C** : 2 bus I2C
7. **SPI** : 4 SPI (Serial Peripheral Interface)
8. **UART** : 3 UART (Universal Asynchronous Receiver/Transmitter)
9. **I2S** : 2 bus I2S (Inter-IC Sound)
10. **CAN** : 1 bus CAN (Controller Area Network)
11. **SD/MMC Host** : 1 interface SD/MMC



**Gambar 2.14 Datasheet ESP32**

(https://fity.club/)

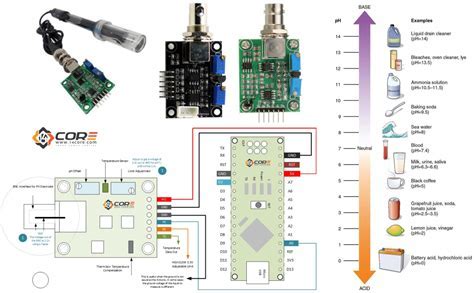
2.6. Sensor dan Modul pH 4502C

Sensor dan modul pH 4502C digunakan untuk mengukur tingkat keasaman atau kebasaan suatu larutan. Sensor ini terdiri dari probe pH yang berfungsi mendeteksi ion hidrogen (H+) dalam larutan. Potensial yang dihasilkan oleh elektroda kaca diteruskan ke modul pH 4502C yang mengubah sinyal potensial menjadi tegangan yang dapat dibaca oleh mikrokontroler atau perangkat lainnya. (Pratama, I., & Nugroho, D., 2019).

Sensor pH mengukur aktivitas ion hidrogen dalam larutan dengan menggunakan elektroda khusus. Prinsip kerjanya melibatkan pengukuran perbedaan potensial listrik antara elektroda pengukur dan elektroda referensi. Perbedaan potensial ini berbanding lurus dengan konsentrasi ion hidrogen dalam larutan, yang kemudian dikonversi ke nilai pH (Yudhana et al., 2018).

Apa yang diukur oleh sensor pH :

1. Konsentrasi ion hidrogen (H+) dalam larutan
2. Tingkat keasaman atau kebasaan larutan
3. Perubahan pH dalam sistem dinamis
4. Kualitas air dalam berbagai aplikasi



**Gambar 2.15 Sensor pH dan Modul Sensor pH 4502C**

(https://www.linkedin.com/)

Keterangan pin pada modul sensor ph :

* + 1. **TO** : Pin ini biasanya digunakan untuk mengeluarkan sinyal yang berhubungan dengan suhu atau temperatur.
    2. **DO** : Pin ini mengeluarkan sinyal digital yang menunjukkan apakah pH larutan berada di atas atau di bawah nilai ambang batas tertentu yang telah diatur.
    3. **PO** : Pin ini adalah pin output yang mengeluarkan tegangan analog yang sebanding dengan nilai pH larutan.
    4. **Gnd :** Pin ini adalah pin ground yang harus dihubungkan ke ground dari sistem (mikrokontroler atau sumber daya).
    5. **Gnd** : Pin ini adalah pin ground yang harus dihubungkan ke ground dari sistem (mikrokontroler atau sumber daya).
    6. **VCC** : Pin ini adalah pin daya yang digunakan untuk memberikan tegangan ke sensor, biasanya menggunakan 5V DC.
    7. **POT 1** : Batas pengukuran temperatur
    8. **POT 2** : Batas pengukuran ph

Sensor dan Modul pH 4502C dirancang untuk mengukur rentang pH dari 0 hingga 14, mencakup spektrum dari sangat asam hingga sangat basa. Sensor ini digunakan dalam berbagai aplikasi seperti akuakultur, hidroponik, dan pemantauan kualitas air, dengan akurasi ±0.1 pH" (Ramadhani, A., & Sari, N., 2020).



**Gambar 2.16 Tingkat Keasaman pH**

(https://devhiariska.blogspot.com/)

Keakuratan pengukuran pH sangat penting dalam banyak aplikasi. Oleh karena itu, kalibrasi rutin sensor pH menggunakan larutan buffer standar sangat diperlukan untuk memastikan pembacaan yang akurat dan konsisten (Purba et al., 2020).

**2.6.1. Cara Kerja Sensor dan Modul pH 4502C**

Kalibrasi sensor pH melibatkan penyesuaian dua parameter utama: slope dan intercept. Slope menunjukkan tingkat perubahan tegangan terhadap perubahan satuan pH, sementara intercept adalah tegangan yang dihasilkan pada pH netral (pH 7). Kedua parameter ini penting untuk mengkonversi sinyal potensial dari elektroda pH menjadi nilai pH yang akurat (Rahmawati, E., & Putra, A., 2019).

Sebelum menentukan nilai slope dan intercept, proses kalibrasi sangat dibutuhkan untuk mendapatkan nilai slope dan intercept. Proses kalibrasi nilai pH bisa menggunakan alternatif bubuk kalibrasi pH



**Gambar 2. Bubuk Kalibrasi Sensor pH**

([https://s3.bukalapak.com/](https://s3.bukalapak.com/img/815174966/large/Set_Bubuk___Serbuk_Kalibrasi_pH_Meter___pH_Buffer_Calibratio.jpg))

Setelah mendapatkan sampel dari bubuk pH, berikut adalah rumus yang digunakan untuk menentukan nilai slope :

Keterangan :

Setelah mendapatkan nilai slope selanjutnya menghitung nilai intercept yang dibituhkan, berikut adalah rumus yang digunakan :

Keterangan :

Setelah mendapatkan nilai slope dan intercept, nilai pH dapat diketahui dengan mengguanka rumus berikut :

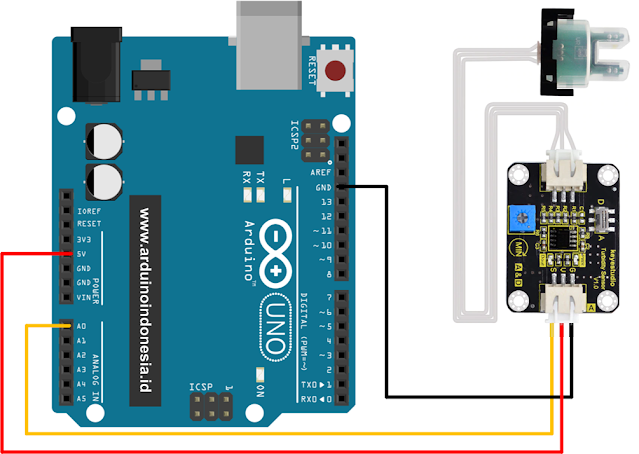
Keterangan :

2.7. Sensor dan Modul Turbidity SEN0189

Sensor turbidity, atau sensor kekeruhan, adalah perangkat yang digunakan untuk mengukur tingkat kekeruhan dalam cairan, terutama air. Turbiditas adalah ukuran kejernihan atau kekeruhan cairan yang disebabkan oleh partikel tersuspensi yang tidak terlihat oleh mata telanjang. Semakin banyak partikel dalam cairan, semakin tinggi turbiditas atau kekeruhannya. Sensor turbidity mengukur intensitas cahaya yang ditransmisikan dan dihamburkan oleh partikel tersuspensi dalam cairan. Prinsip kerjanya melibatkan sumber cahaya (biasanya LED inframerah) dan fotodetektor yang mengukur intensitas cahaya yang diterima setelah melewati sampel cairan (Sabiq & Budisejati, 2017).

Apa yang sensor turbidity ukur :

1. Intensitas cahaya yang ditransmisikan melalui cairan
2. Intensitas cahaya yang dihamburkan oleh partikel tersuspensi
3. Konsentrasi partikel tersuspensi dalam cairan
4. Tingkat kekeruhan air dalam satuan NTU (Nephelometric Turbidity Units) atau FNU (Formazin Nephelometric Units)

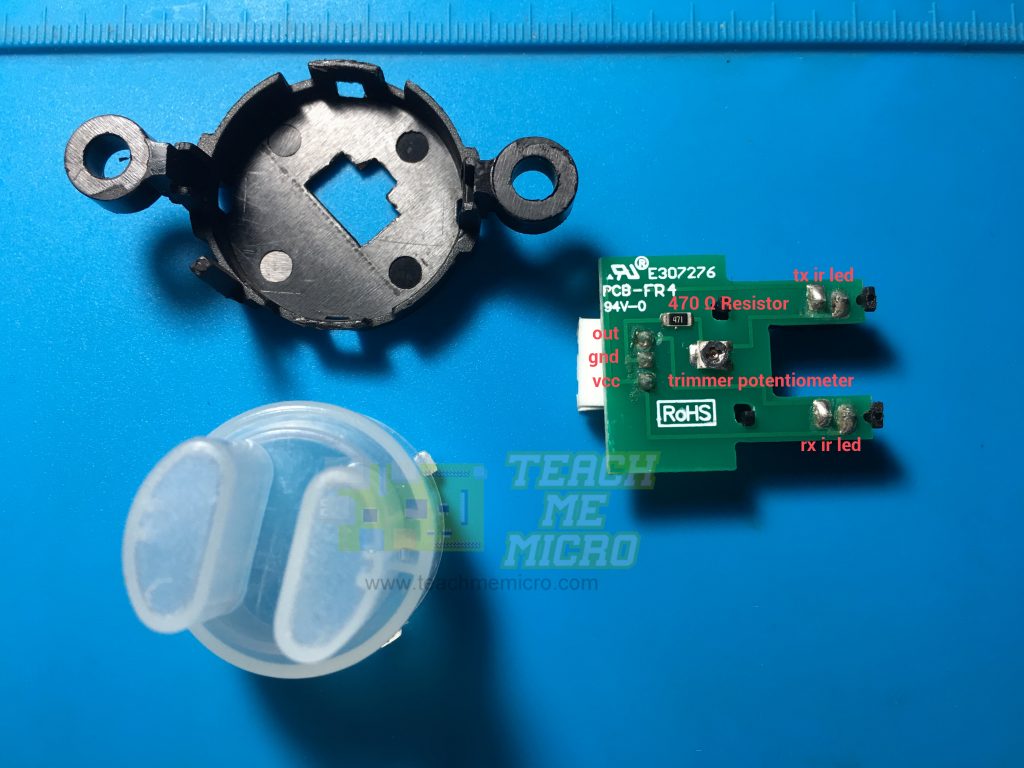


**Gambar 2.17 Sensor Turbidity dan Modul SEN0189**

(<https://www.arduinoindonesia.id/>)

Keterangan pin yang digunakan :

* 1. **Signal** : Pin ini mengeluarkan sinyal analog yang sebanding dengan tingkat kekeruhan air. Sinyal ini dapat dibaca oleh ADC (Analog-to-Digital Converter) dari mikrokontroler untuk mendapatkan nilai digital yang mewakili tingkat kekeruhan.
  2. **VCC** : Pin ini adalah pin daya yang digunakan untuk memberikan tegangan ke sensor biasanya 5V DC
  3. **Gnd :** Pin ini adalah pin ground yang harus dihubungkan ke ground dari sistem (mikrokontroler atau sumber daya).
     1. **Cara Kerja Sensor dan Modul Turbidity SEN0189**



**Gambar 2. Sensor Turbidity**

(<https://www.teachmemicro.com/>)

Kalibrasi sensor turbidity sangat penting untuk memastikan akurasi pengukuran. Biasanya, kalibrasi dilakukan menggunakan standar formazin atau standar kekeruhan lainnya yang telah diketahui nilainya (Ihsanto & Hidayat, 2020).

Dalam mencari nilai kekeruhan diperlukan beberapa persamaan yang diibutuhkan dan juga sama seperti menentukan nilai pH, nilai turbidity memerlukan kalibrasi terlebih dahulu. Kalibrasi bisa mengguanakan 2 contoh air (air keruh dan air jernih). Berikut adalah rumus yang digunakan utnuk menentukan niali turbidity (kekeruhan) :

Keterangan :

2.8. Sensor Ultrasonik

HC-SR04 adalah modul sensor ultrasonik yang terdiri dari pemancar ultrasonik (transmitter), penerima ultrasonik (receiver), dan rangkaian kontrol. Sensor ini bekerja dengan mengirimkan gelombang ultrasonik berfrekuensi 40 kHz dan kemudian menunggu pantulannya. Dengan mengukur waktu antara pengiriman dan penerimaan gelombang, sensor dapat menghitung jarak ke objek (Andrianto et al., 2020).

Apa yang diukur Sensor Ultrasonik HC-SR04 :

1. Jarak antara sensor dan objek (dalam rentang 2 cm hingga 400 cm)
2. Waktu tempuh gelombang ultrasonik (echo time)
3. Keberadaan objek dalam jangkauan sensor

Spesifikasi teknis Sensor Ultrasonik HC-SR04 :

* Tegangan operasi: 5V DC
* Arus statis: <2mA
* Sudut efektif: <15°
* Jarak pengukuran: 2cm - 400cm
* Resolusi: 0.3cm
* Frekuensi ultrasonik: 40kHz



**Gambar 2.17 Sensor Ultrasonik HC – SR04**

(<https://www.mecanicaindustrial.com.br/>)

Pin pada sensor Ultrasonik HC-SR04 :

1. **VCC** : Pin ini adalah pin daya yang digunakan untuk memberikan tegangan ke sensor, biasanya menggunakan tegangan 5V DC.
2. **Trig** : Pin ini digunakan untuk memicu sensor untuk mulai mengirimkan sinyal ultrasonik. Ketika diberikan sinyal HIGH (5V) selama setidaknya 10 mikrodetik, sensor akan mengirimkan sinyal ultrasonik.
3. **Echo** : Pin ini adalah pin keluaran yang digunakan untuk mengirimkan durasi waktu ketika sinyal ultrasonik kembali setelah mengenai objek.
4. **Gnd** : Pin ini adalah pin ground yang harus dihubungkan ke ground dari sistem (mikrokontroler atau sumber daya).
   * 1. **Cara Kerja Sensor Ultarsonik HC-SR04 :**
   1. Mengukur jarak

Pin Trigger akan memancarkan sinyal ultrasonik selama 10µs, dan pin Echo menerima kembali sinyal tersebut setelah dipantulkan. Kemudian kita asumsikan nilai dari sinyal ultrasonik bolak-balik tersebut disimpan dalam variabel Durasi. Selanjtnya untuk menentukan jarak sebenarnya menggunakan rumus :

Keterangan :

Jarak = Nilai yang dicari (cm)

Durasi = Nilai bolak-balik dari sinyal ultrasonik yang dipancarkan

0.034 = Nilai untuk mewakili kecepatan suara dalam udara.

2 = Nilai dari dipancarkan dan ditangkap (bolak-balik)

2. Mengukur Tinggi air

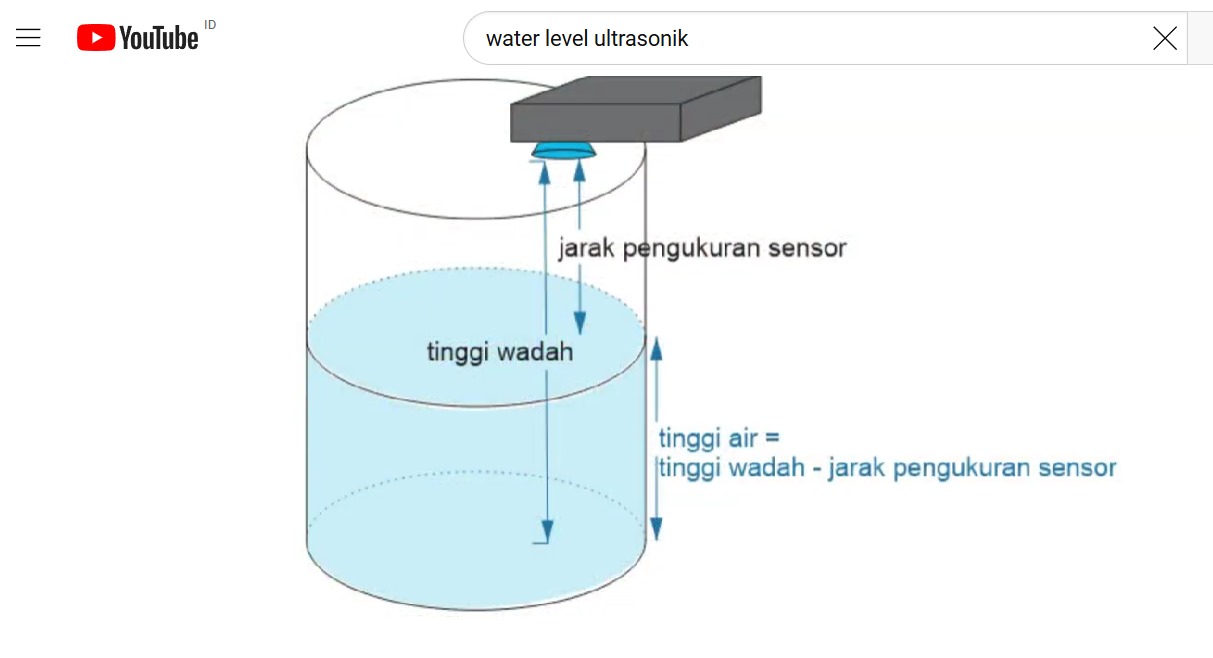
Setelah mengetahui jarak yang dibutuhkan dalam pengukuran, selanjutnya menentukan tinggi air pada aquarium. Tinggi air dapat dihitung dengan rumus berikut :

Keterangan :

Tinggi air = Nilai yang dicari (cm)

Tinggi wadah = Nilai jarak dasar wadah dengan sensor ultrasonik

Jarak = Nilai dari perhitungan sebelumnya



**Gambar 2.**

HC-SR04 sering diintegrasikan dengan mikrokontroler seperti Arduino atau ESP32 untuk berbagai aplikasi. Keunggulannya termasuk harga yang terjangkau, kemudahan penggunaan, dan kompatibilitas dengan berbagai platform pengembangan (Prawiroredjo & Asteria, 2018).

2.9. Motor Servo

Motor servo adalah perangkat elektromekanis yang dapat diatur untuk bergerak atau mempertahankan posisi tertentu. Berbeda dengan motor DC biasa, motor servo memiliki kemampuan untuk dikontrol dalam hal posisi sudut shaft-nya. Motor servo biasanya terdiri dari motor DC, gearbox, potensiometer sebagai sensor posisi, dan rangkaian kontrol (Andrianto et al., 2019).

Karakteristik umum motor servo:

* Rentang sudut putar: biasanya 0-180 derajat (untuk servo standar)
* Tegangan operasi: umumnya 4.8-6V
* Torsi: bervariasi tergantung ukuran dan jenis servo
* Kecepatan: biasanya diukur dalam detik/60 derajat



**Gambar 2.18 Motor Servo**

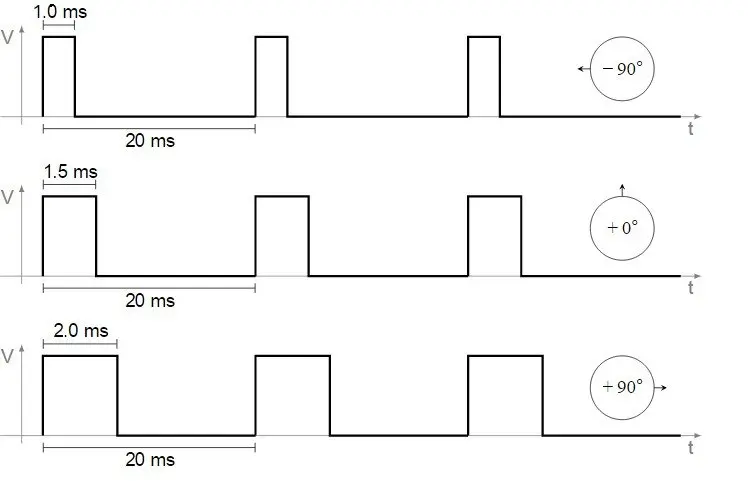
(<https://uelectronics.com/>)

Pin pada motor servo :

1. **Signal :** Pin ini digunakan untuk menerima sinyal kontrol dari mikrokontroler atau sumber sinyal PWM (Pulse Width Modulation).
2. **VCC :** Pin ini adalah pin daya yang digunakan untuk memberikan tegangan ke motor servo, tegangan yang dibutuhkan 5V DC
3. **Gnd :** Pin ini adalah pin ground yang harus dihubungkan ke ground dari sistem (mikrokontroler atau sumber daya).

**2.9.1 Cara Kerja Motor Servo :**

Motor Servo dikontrol menggunakan PWM (Pulse Width Modulation), PWM digunakan untuk menentukan posisi sudut motor servo dengan mengubah lebar pulse dari sinyal kontrol. Sinyal PWM terdiri dari pulsa dengan frekuensi tetap (misalnya 50 Hz, yang berarti setiap detik ada 50 siklus), di mana lebar setiap pulsa bervariasi. Lebar pulsa menentukan seberapa lama sinyal berada pada level tinggi (ON) dalam satu siklus. Untuk motor servo, lebar pulse ini berkisar antara 1 ms (milidetik) hingga 2 ms dalam satu siklus 20 ms (50 Hz). Lebar pulsa yang lebih pendek biasanya menghasilkan posisi satu arah (misalnya sudut 0 derajat), sedangkan lebar pulsa yang lebih panjang menghasilkan posisi arah lain (misalnya sudut 180 derajat).



**Gambar 2. PWM Pada Motor Servo**

(https://www.plcgurus.net/)

2.10. Pompa Aquarium

Pompa akuarium adalah alat mekanis yang menggunakan motor listrik untuk menggerakkan impeller (baling-baling), yang kemudian menciptakan aliran air. Pompa ini umumnya dirancang untuk penggunaan terus-menerus dan dapat dicelupkan langsung ke dalam air (submersible) atau ditempatkan di luar akuarium (external) (Kusrini et al., 2019).



**Gambar 2.19 Pompa Aquarium**

(<https://makassarhobi.com/>)

**2.10.1. Cara Kerja Pompa Aquarium**

Cara kerja yang terjadi pada pompa biasanya tergantung pompa yang digunakan. Pompa yang digunakan pada sistem ini yaitu pompa internal (Pompa Submersible), yang dimana pompa dipasang didalam aquarium. Namun secara umum semua pompa memiliki cara kerja yang sama. Pertama air dalam aquarium akan masuk kedalam pompa. Kedua setelah pompa diberikan tegangan (biasanya 220V), motor yang ada dalam pompa akan berputar dan menggerakan impeller (rotor), kemudian impeller ini mendorong air yang ada didalam pompa dan menciptakan tekanan sehingga air terdorong masuk ke pipa pada pompa air menuju saringan (filter) dan setelah air keluar dari filter, air akan kembali jernih. Proses ini berulang-ulang, sehingga menciptakan sirkulasi yang merata. Ini membantu mendistribusikan oksigen dan nutrisi di seluruh akuarium serta mencegah pembentukan area stagnan.

2.11. Modul Step-Up MT3608

MT3608 adalah sebuah konverter DC-DC step-up yang menggunakan teknik switching untuk meningkatkan tegangan input menjadi tegangan output yang lebih tinggi. Modul ini memungkinkan pengguna untuk mengkonversi sumber tegangan rendah (misalnya dari baterai) menjadi tegangan yang lebih tinggi untuk menjalankan perangkat elektronik (Pratama et al., 2020).

Karakteristik utama MT3608:

1. Tegangan input: 2V - 24V
2. Tegangan output: hingga 28V (tergantung pada tegangan input)
3. Arus output maksimum: 2A
4. Efisiensi konversi: hingga 93%
5. Frekuensi switching: 1.2 MHz (tetap)



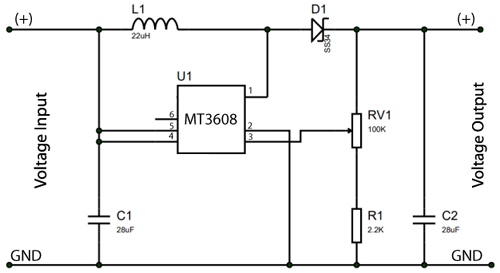
**Gambar 2.20 Modul Step-Up MT3608**

(<https://shop.stemlabs.edu.lk/>)

Pin pada modul step-up MT3608 :

1. **Vin + :** Pin ini adalah pin input positif yang menerima tegangan dari sumber daya.
2. **Vin - :** Pin ini adalah pin ground atau negatif dari sumber daya input.
3. **Vout + :** Pin ini adalah pin output positif yang memberikan tegangan yang telah ditingkatkan.
4. **Vout - :** Pin ini adalah pin ground atau negatif dari output tegangan.

**2.11.1. Cara Kerja Modul Step-UP MT3608**



**Gambar 2. Skema Modul Step-Up MT3608**

(https://components101.com/)

Berdasarkan *gambar 2.* Modul Step-Up MT3608 memiliki beberapa komponen yang mendukung dalan menaikan tegangan, sehingga mendpatkan tegangan masukan untuk komponen yang cukup, berikut adalah tahapan cara kerja Modul Step-Up MT3608. Pertama tegangan masukan pada modul Step-Up MT3608 lebih rendah dari tegangan yang dibutuhkan untuk komponen yang ingin digunakan. Misalnya dibutuhkan 5V untuk ESP32. Kedua, setelah tegangan masukan diberikan, transistor di dalam IC MT3608 akan dihidupkan, kemudian arus akan mengalir melalui induktor dan transistor ke ground. Selanjutnya induktor menyimpan energi dalam bentuk medan magnet. Ketiga, ketika transistor di dalam IC MT3608 dimatikan, arus dari induktor tidak dapat mengalir melalui transistor lagi dan dioda Schottky mulai mengalirkan arus. Medan magnet yang disimpan di induktor mulai runtuh, melepaskan energi yang disimpan sebagai tegangan yang lebih tinggi. Keempat, tegangan yang dihasilkan dari induktor yang runtuh mengalir melalui dioda Schottky dan mengisi kapasitor output. Untuk mengatur

2.12. Modul Step-Down LM259

LM2596 adalah sebuah integrated circuit (IC) yang digunakan dalam modul konverter DC-DC step-down. Modul ini berfungsi untuk menurunkan tegangan input DC menjadi tegangan output DC yang lebih rendah dengan efisiensi tinggi. Modul LM2596 menggunakan teknik switching untuk mengkonversi tegangan, yang membuatnya lebih efisien dibandingkan dengan regulator linear (Andrianto et al., 2019).

Karakteristik utama LM2596:

1. Tegangan input: 3.2V - 40V
2. Tegangan output: 1.25V - 35V (dapat diatur)
3. Arus output maksimum: 3A
4. Efisiensi konversi: hingga 92%
5. Frekuensi switching: 150 kHz (tetap)



**Gambar 2.21 Modul Step-Down LM259**

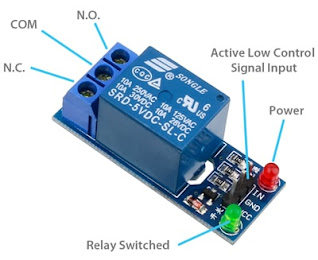
(<https://www.benselectronics.nl/>)

Keterangan Pin :

* 1. **Vin + :** Pin ini adalah pin input positif yang menerima tegangan dari sumber daya.
  2. **Vin - :** Pin ini adalah pin ground atau negatif dari sumber daya input.
  3. **Vout + :** Pin ini adalah pin output positif yang memberikan tegangan yang telah diturunkan.
  4. **Vout - :** Pin ini adalah pin ground atau negatif dari output tegangan.

2.13. Relay 1 Channel

Relay 1 channel adalah komponen elektromekanikal yang terdiri dari kumparan elektromagnet dan satu set kontak saklar. Ketika kumparan diberi energi, medan magnet yang dihasilkan menarik tuas yang mengubah posisi kontak saklar. Ini memungkinkan relay untuk menghubungkan atau memutuskan rangkaian listrik terpisah dengan arus dan tegangan yang lebih tinggi menggunakan sinyal kontrol bertenaga rendah (Andrianto et al., 2020).



**Gambar 2.22 Modul Relay 1 Channel**

(<https://www.aldyrazor.com/>)

Keterangan Pin :

* 1. **NC :** Pin ini adalah pin yang biasanya tertutup (terhubung ke COM) saat relay tidak aktif.
  2. **COM :** Pin ini adalah pin umum yang terhubung ke kontak NO atau NC tergantung pada status relay (aktif atau nonaktif).
  3. **NO :** Pin ini adalah pin yang biasanya terbuka (tidak terhubung) saat relay tidak aktif.
  4. **IN :** Pin ini digunakan untuk menerima sinyal kontrol dari mikrokontroler atau sumber sinyal lainnya.
  5. **GND :** Pin ini adalah pin ground yang harus dihubungkan ke ground dari sistem (mikrokontroler atau sumber daya).
  6. **VCC :** Pin ini adalah pin daya yang digunakan untuk memberikan tegangan ke sirkuit kontrol relay.

2.14. Baterai

Batu baterai adalah sel elektrokimia yang terdiri dari elektroda positif (katoda), elektroda negatif (anoda), dan elektrolit. Ketika baterai dihubungkan ke rangkaian, terjadi reaksi kimia yang menghasilkan aliran elektron, menciptakan arus listrik (Linden & Reddy, 2019).



**Gambar 2.23 Batu Baterai**

(<https://www.jakartanotebook.com/>)

2.15. Switch

Switch adalah komponen yang memiliki dua keadaan: terbuka (off) dan tertutup (on). Ketika switch tertutup, arus listrik dapat mengalir melaluinya; ketika terbuka, aliran listrik terputus. Switch dapat dioperasikan secara manual atau otomatis, tergantung pada jenisnya (Malvino & Bates, 2020).



**Gambar 2.24 Switch**

(<https://makerselectronics.com/>)

2.16. Powerbank

Powerbank adalah baterai eksternal yang dilengkapi dengan rangkaian elektronik untuk mengatur pengisian dan pengeluaran daya. Perangkat ini umumnya menggunakan baterai lithium-ion atau lithium-polymer karena kepadatan energi yang tinggi dan kemampuan pengisian ulang yang baik (Zhang et al., 2019).



**Gambar 2.25 Powerbank**

(<https://www.discoazul.com/>)

2.17. Kabel Micro USB

Kabel USB (Universal Serial Bus) adalah perangkat keras antarmuka yang menghubungkan komputer dengan periferal eksternal, memungkinkan transfer data dan daya listrik melalui satu konektor standar. Desainnya yang universal memungkinkan berbagai jenis perangkat untuk terhubung dan berkomunikasi dengan komputer host, menjadikannya solusi konektivitas yang sangat serbaguna dalam komputasi modern. (Johnson & Lee, 2021).



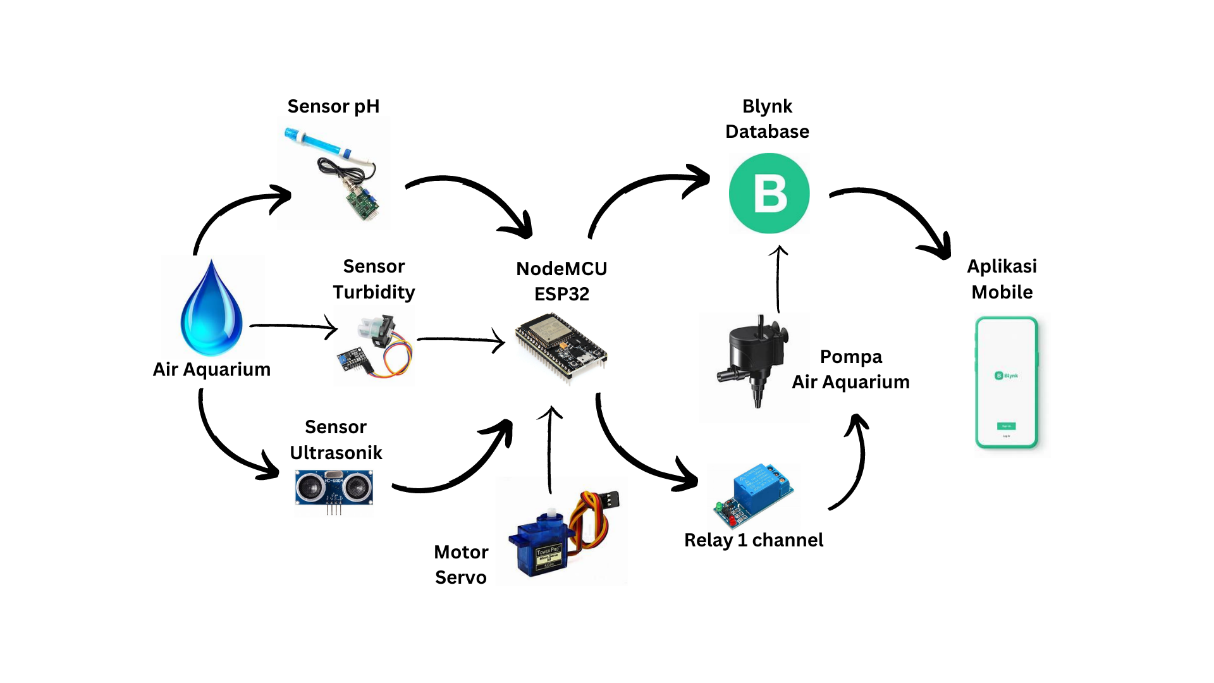
**Gambar 2.26 Kabel Micro USB**

(<https://www.walmart.com/>)

**3. PERANCANGAN DAN IMPLEMENTASI**

**3.1. Gambaran Umum Kerja Sistem**

Sistem aquarium ini merupakan alat yang dirancang untuk memudahkan pemeliharaan ikan hias khususnya ikan mas koki agar dapat dipantau secara otomatis. Dalam pengimplementasian alat ini, digunakan 2 sensor untuk memantau kualitas air yang ideal untuk ikan mas koki yaitu, sensor pH dan sensor Turbidity, kedua sensor ini digunakan sebagai trigger untuk mengatifkan pompa dan memfilter air agar kembali jernih. Selain memantau kualitas air terdapat pula sensor ultrasonik untuk memantau ketinggian air pada aquarium serta motor servo untuk memberikan makanan ikan secara otomatis, yang dapat dikendalikan dengan penjadwalan melalui aplikasi Blynk



***Gambar* 3.1 Gambaran Umum Sistem Aquarium Pintar Bberbasis Blynk**

**3.2. Perancangan Alat**

Dalam pembuatan alat ini berdasarkan analisa dan tahapan perencanaan yang telah dilakukan, diperlukan pembuatan perancangan dari segi perangkat keras maupun perangkat lunak

1. **Perangkat keras**

Perancangan perangkat keras yang dibutuhkan pada Aquarium Pintar berbasis IoT menggunakan Blynk

* NodeMCU ESP32 : digunakan sebagai pemrosesan data pada sistem yang digunakan
* Sensor pH 4502C : digunakan untuk mendeteksi nilai pH pada air
* Sensor Turbidity SEN0189 : digunakan sebagai pendeteksi tingkat kekeruhan air pada aquarium
* Sensor Ultrasonik HC-SR04 : digunakan untuk mendeteksi ketinggian air pada aquarium
* Motor Servo : berfungsi untuk memberikan makanan ikan secara otomatis maupun manual
* Modul Step-up MT3608 : berfungsi untuk menaikan tegangan output dan memberikan tegangan yang sesuai kebutuhan komponen
* Modul Step-down LM259 : berfungsi untuk menurunkan tegangan output dan memberikan tegangan sesuai dengan kebutuhan komponen
* Fan : digunakan untuk mendinginkan komponen perangkat keras yang digunakan
* Relay : berfungsi sebagai saklar dan menjalankan logika yang diberikan untuk menjalankan pompa dan mengganti cadangan daya dari adapter ke powerbank
* Switch : sebagai inputan ke relay untuk mengganti cadangan daya
* Baterai 3.25 V : berfungsi sebagai daya untuk menjalankan relay yang mengatur cadangan daya
* Pompa aquarium : digunakan untuk menyaring air ketika sudah keruh atau pH naik

1. **Perangkat Lunak**

Perancangan perangkat lunak yang dibutuhkan pada Aquarium Pintar berbasis IoT menggunakan Blynk

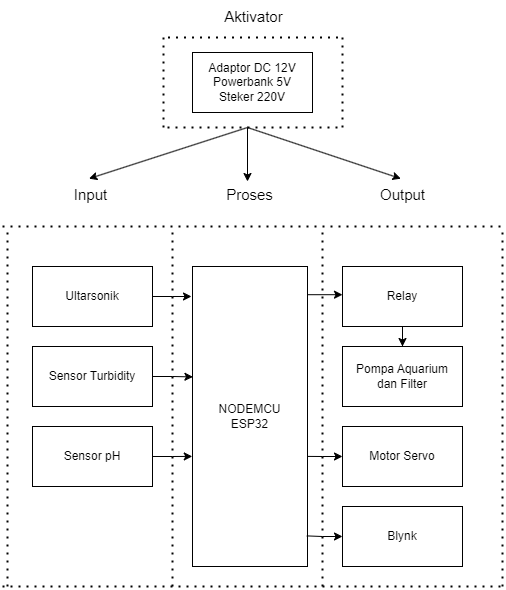
* Arduino IDE

Arduino IDE digunakan untuk membuat, mengubah dan menghapus code program yang ingin digunakan untuk mengkontrol suatu projek berbasis mikrokontroler. Arduino IDE memiliki banyak library yang dapat digunakan untuk komponen yang ingin digunakan pada suatu proyek, serta memiliki fitur untuk mengunggah kode melalui koneksi USB. Selain itu, Arduino IDE memiliki Serial Monitor untuk debugging.

* Blynk

Blynk adalah platform IoT (Internet of Things) yang memungkinkan pengguna untuk mengontrol dan memantau perangkat keras melalui aplikasi mobile. Dengan Blynk, pengguna dapat membuat antarmuka pengguna grafis (GUI) yang interaktif untuk perangkat IoT tanpa harus menulis banyak kode.

**3.2.1. Perancangan Perangkat Keras**

****

***Gambar* 3.2 Blok Diagram**

Rancangan sistem Aquarium Pintar dapat dilihat melalui blok diagram pada ***Gambar 3.2.*** Blok diagram diperlukan dalam perancangan sistem karena memberikan gambaran visual yang jelas dan sederhana mengenai komponen-komponen utama dan hubungan antar komponen dalam sistem. Berikut adalah penjelasan masing-masing blok yang digunakan :

**3.2.1.1. Blok Aktivator**

Pada blok aktivator terdapat Adaptor DC 12V sebagai sumber daya utama untuk komponen yang digunakan selain pompa aquarium. Modul Step-Down LM259 digunakan untuk menurunkan tegangan masukan adaptor dari 12V menjadi 5V sesuai dengan kebutuhann tegangan masukan ESP32. Powerbank dengan keluaran 5V digunakan sebagai sumber daya cadangan apabila terjadi pemadaman listrik. Dan steker 220V merupakan daya untuk menjalankan pompa aquarium. Modul Step-Up MT3608 pertama digunakan untuk menaikan voltase dari relay pertama, karena keluaran dari relay pertama kurang dari 5V, sedangkan ESP32 membutuhkan daya masukan sebesar 5V agar semua komponen bisa berfungsi sebagaimana mestinya.

**3.2.1.2. Blok Input**

Pada blok input terdapat sensor ultrasonik sebagai masukan ketinggian air pada aquarium. Perhitungan tinggi air pada aquarium dapat dihitung dengan cara *Tinggi air = tinggi wadah – jarak pengukuran sensor.* Pin *trigger* memancarkan gelombang ultrasonik, kemudian gelombang tersebut terpantul kembali ke sensor dan di tangkap oleh pin *echo.* Sensor turbidity digunakan untuk mendeteksi tingkat kekeruhan yang ada pada aquarium, dan sensor pH digunakan untuk mendeteksi tingkat keasaman air / pH yang terkandung dalam air tersebut. Ketiga sensor tersebut akan membaca keadaan air aquarium dan mengirimkan data ke mikrokontroller ESP32 untuk diproses dan menghasilkan output pada aplikasi blynk yang sudah terintegrasi secara realtime.

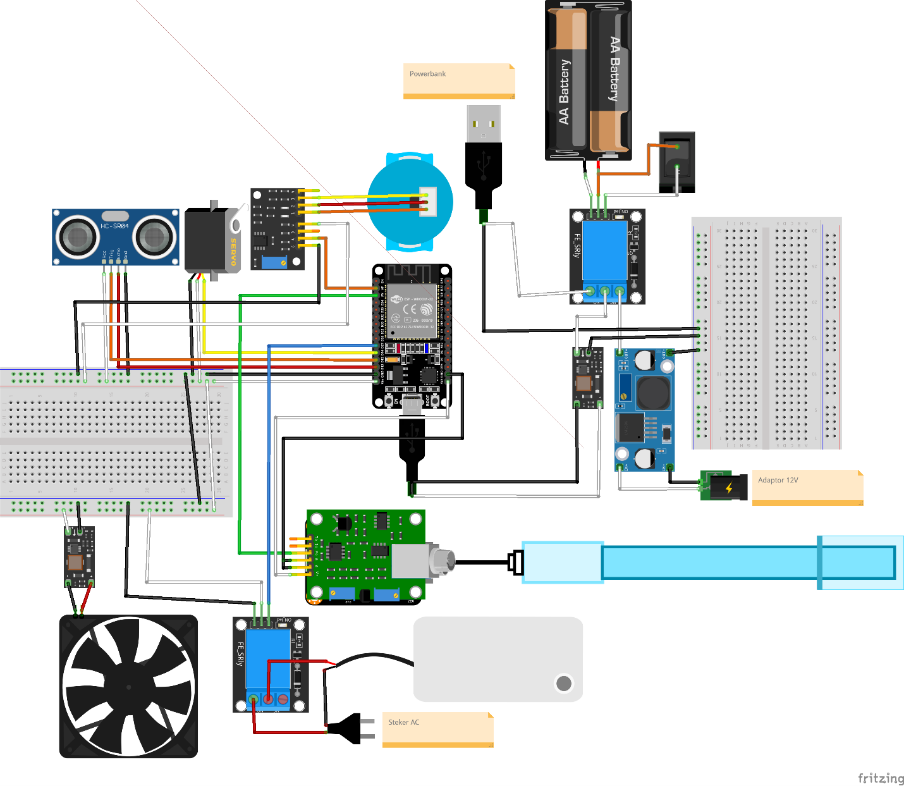
**3.2.1.3. Blok Proses**

Pemrosesan yang dilakukan pada alat ini adalah mikrokontroller NodeMCU ESP32. Pada blok ini semua data yang didapatkan dari setiap sensor akan di eksekusi sesuai code perintah yang sudah di upload pada mikrokontroller ini. NodeMCU ESP32 sudah terintegrasi secara realtime dengan aplikasi blynk, sehingga setelah data diproses maka akan diteruskan ke blok output.

**3.2.1.4. Blok Output**

Sedangkan pada blok output terdapat relay untuk mengkontrol pompa aktif atau mati, pompa akan aktif jika memenuhi kondisi yang sudah ditentukan pada mikrokontroller tersebut. Selain pompa, motor servo memberikan output berupa makanan ikan sesuai jadwal yang diatur pada aplikasi blynk dan fan sebagai keluaran untuk mendinginkan komponen yang digunakan pada sistem ini. Semua data

Berdasarkan blok diagram pada gambar 3, rangkaian skematik sistem aquarim pintar ini dapat diuraikan sebagai berikut :



**Gambar 3.3 Skematik Rangkaian**

Pada **Gambar 3.3** terdapat rangkaian skematik dari sistem aquarium pintar ini, untuk lebih memahami lebih detail terkait komponen apa saja yang digunakan dan bagaimana setaip komponen saling terhubung, dapat dilihat pada tabel berikut :

**Tabel 3.1 Komponen dan Pin**

|  |  |  |  |
| --- | --- | --- | --- |
| KOMPONEN | PIN KOMPONEN | TERHUBUNG KE | |
| PIN | KOMPONEN |
| NodeMCU ESP32 | + 5v | Positif | Breadboard |
| Gnd | Negatif | Breadboard |
| D12 | Trig | Ultrasonik |
| D13 | Echo | Ultrasonik |
| D14 | Sinyal | Motor Servo |
| D27 | Common | Relay 2 |
| Vn | Po | Modul pH |
| Vp | Analog | Modul Turbidity |
| Motor Servo | Sinyal | D14 | NodeMCU ESP32 |
| Vcc | Posiitif | Breadboard |
| Gnd | Negatif | Breadboard |
| Ultrasonik HC-SR04 | Vcc | Positif | Breadboard |
| Echo | D13 | NodeMCU ESP32 |
| Trig | D12 | NodeMCU ESP32 |
| Gnd | Negatif | Breadboard |
| Modul Sensor pH 4502C | Vcc | 3.3v | NodeMCU ESP32 |
| Gnd | Gnd | NodeMCU ESP32 |
| Po | Vn | NodeMCU ESP32 |
| To | - | - |
| Do | - | - |
| Modul Sensor Turbidity SEN0189 | Gnd | Negatif | Breadboard |
| Analog | Vp | NodeMCU ESP32 |
| Digital | None | None |
| Vcc | Positif | Breadboard |
| 1 | Vcc | Sensor Turbidity |
| 2 | Sinyal | Sensor Turbidity |
| 3 | Gnd | Sensor Turbidity |
| 4 | - | - |
| Sensor Turbidity | Vcc | 1 | Modul sensor Turbidity SEN0189 |
| Sinyal | 2 | Modul sensor Turbidity SEN0189 |
| Gnd | 3 | Modul sensor Turbidity SEN0189 |
| Relay 1 | In | In | Switch |
| Gnd | Negatif | Breadboard |
| Vcc | Vcc | Switch |
| No | Powerbank | Modul Step-up 1 |
| C | Vin + | Modul Step-up 1 |
| Nc | Vout + | Modul Step-down |
| Switch | Vcc | Vcc | Relay 1 |
| In | In | Relay 1 |
| Baterai | Positif (+) | * Vcc * Vcc | * Switch * Relay 1 |
| Negatif (-) | Gnd | Relay 1 |
| Modul Step-up 1 | Vin + | C | Relay 1 |
| Vin - | Negatif | Breadboard |
| Vout + | Kabel USB | NodeMCU ESP32 |
| Vout - | Kabel USB | NodeMCU ESP32 |
| Modul Step-down | Vin + | Positif adapter | Relay 1 |
| Vin - | Negatif adapter | Adapter |
| Vout + | Nc | Relay 1 |
| Vout - | Negatif | Breadboard |
| Relay 2 | In | D27 | NodeMCU ESP32 |
| Vcc | Positif | Breadboard |
| Gnd | Negatif | Breadboard |
| Nc | - | - |
| C | Positif Pompa | Pompa |
| No | Positif Steker | Steker |
| Modul Step-up 2 | Vin + | Positif | Breadboard |
|  | Vin - | Negatif | Breadboard |
|  | Vout + | Negatif Fan | Fan |
|  | Vout - | Positif Fan | Fan |

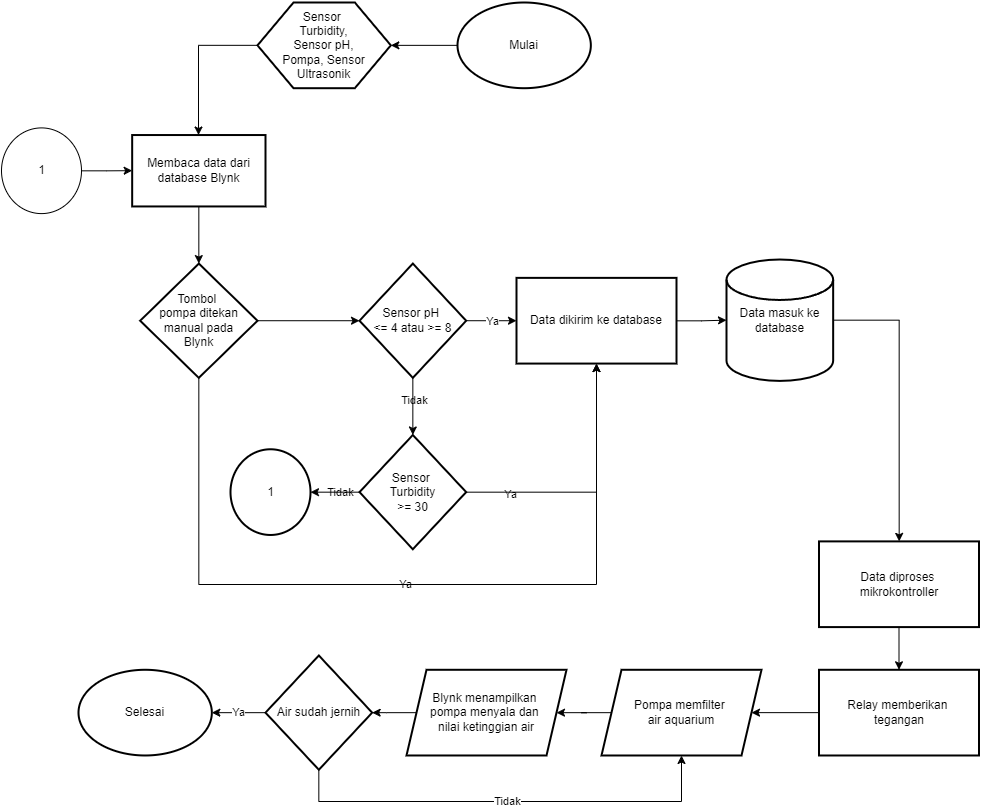
**3.2.2. Perancangan Perangkat Lunak**

Dalam pembuatan sistem sesuai skematik diatas, code program diperlukan untuk mengendalikan setiap komponennya. Bahasa pemrograman yang digunakan yaitu bahasa C dan akan dijelaskan alur program untuk menghasilkan ouput dalam flowchart. Berikut adalah flowchart dari komponen yang digunakan selain motor servo, karena motor servo tidak berkaitan dengan kondisi air pada aquarium, sehingga memiliki flowchart sendiri.



**Gambar 3.4 Tampilan aplikasi Blynk**

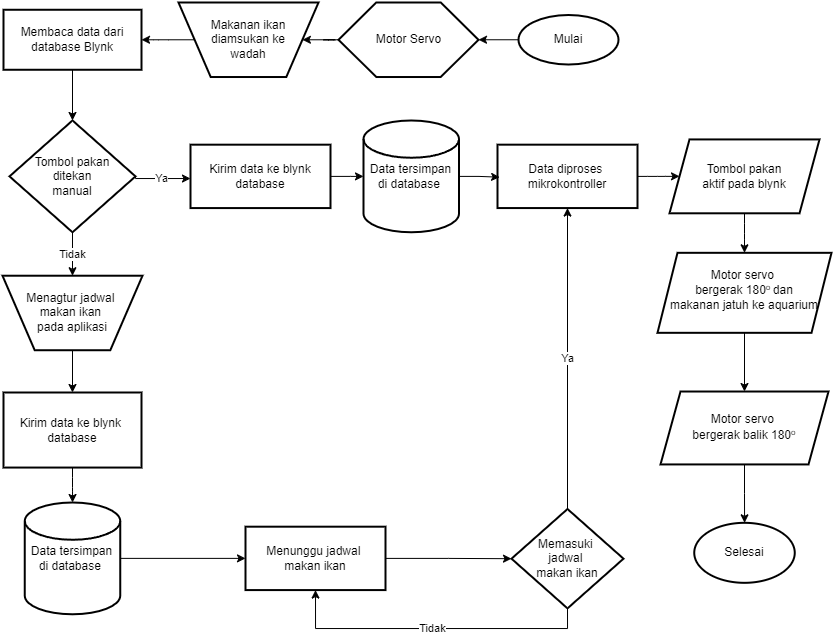
**3.2.2.1 Flowchart**

****

**Gambar 3.5 Flowchart Sensor pH, Sensor Turbidity dan**

**Sensor Ultrasonik**

Berdasarkan *gambar 3.5* dijelaskan bagaimana alur kerja sistem aquarium pintar ini. Dimulai dengan inisialisasi komponen yang digunakan (Sensor pH, sensor turbidity, pompa air dan sensor ultrasonik), pengaktifan mikrokontroller yang digunakan, pendefinisian pin, pendeklarasian variabel yang digunakan dan juga menghubungakan perangkat dengan blynk serta internet. Setelah proses inisialisasi selesai, mikrokontroller akan membaca data awal dari blynk, yang mana data awal itu relay tidak memiliki tegangan sehingga pompa air belum aktif. Kemudian pengguna bisa menekan tombol pompa manual atau tidak pada tampilan aplikasi blynk. Jika pengguna menekan tombol manual *pompa* yang dapat dilihat pada *gambar 3.4*, maka data akan dikirimkan ke blynk database mikrokontroller ESP32 akan memproses data keluaran dari database, sehingga relay diberikan tegangan yang memicu pompa agar menyala. Kemudian jika air tidak keruh maka pompa akan otomatis untuk mati dan program selesai. Namun jika pengguna tidak menekan tombol manual pada aplikasi blynk, sensor sebagai pemicu relay untuk menyalakan pompa. *Kondisi pertama* jika sensor pH bernilai ≤ 4 atau ≥ 8 maka relay diberikan tegangan dan pompa air akan menyala. Jika sensor pH mendeteksi nilai tersebut maka data akan dikirmkan ke database blynk, kemudian akan diproses oleh mikrokontroller ESP32 untuk memberikan tegangan pada relay dan mengaktifkan pompa, pompa akan berhenti jika nilai pH sudah tidak memenuhi kondisi tersebut. *Kondisi kedua* jika sensor turbidity (kekeruhan) bernilai ≥ 30, maka prosesnya akan sama seperti kondisi sensor pH tadi. Jika kedua kondisi tidak terpenuhi maka akan berulang pada proses sebelum pengguna menekan tombol manual pompa pada blynk.



**Gambar 3.6 Flowchart Motor Servo**

Berdasarkan *gambar 3.6* dijelaskan bagaimana alur kerja makanan ikan dari awal sampai akhir. Dimulai dengan inisialisasi komponen yang digunakan (motor servo), pengaktifan mikrokontroller yang digunakan, pendefinisian pin, pendeklarasian variabel yang digunakan dan juga menghubungakan perangkat dengan blynk serta internet. Setelah inisialisasi pengguna memasukan terlebih dahulu makanan ikan pada wadah yang digunakan untuk dikontrol melalui blynk. Selanjutnya mikrokontroller membaca data awal dari database blynk, yang dimana data awal motor servo berlogika 0 (LOW) sehingga motor servo belum bergerak. Kemudian pengguna bisa menekan tombol *pakan* (*gambar 3.4*) secara manual atau melakukan penjadwalan pada aplikasi blynk. Jika pengguna menekan tombol pakan, data akan dikirmkan ke blynk database kemudian diproses oleh mikrokontroller ESP32 dan memberikan logika 1 (HIGH) pada motor servo, sehingga motor servo akan bergerak 180° dan makanan ikan akan jatuh ke aquarium, selanjutnya motor servo akan kembali ke posisi 0° (posisi awal). Namun jika pengguna tidak menekan tombol pakan pada aplikasi blynk, maka pengguna bisa mengatur jadwal kapan makanan ikan tersebut akan diberikan ke ikan. Setelah pengguna mengatur jadwal yang diinginkan untuk memberikan makanan ikan, maka data akan dikirimkan ke blynk database dan proses seterusnya sama dengan proses ketika setelah menekan tombol *pakan* secara manual dari aplikasi blynk.

**3.2.2.2 Analisa Program**

Berdasarkan flowchart pada *gambar 3.5* dan *gambar 3.6* berikut adalah penjelasan setiap baris code untuk menjalankan sistem *AquaSmart Berbasis Blynk IoT* agar sesuai dengan kedua gambar tersebut.



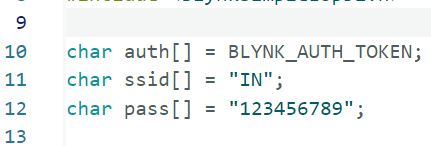
**Gambar 3.7 Menghubungkan Blynk dan Library**

**Yang Dibutuhkan**

Berikut adalah penjelasan dari code / instruksi yang diberikan pada *gambar 3.7 :*

**Tabel 3.2 Menghubungkan Blynk dan Library Yang Dibutuhkan**

|  |  |
| --- | --- |
| **Program** | **Keterangan** |
| #define BLYNK\_TEMPLATE\_ID "TMPL6n5YuDce4"  #define BLYNK\_TEMPLATE\_NAME "AquaSmart"  #define BLYNK\_AUTH\_TOKEN "s75chNlwx1Cx7yjJDr2RLzf77005UFCT" | Baris code ini digunakan untuk mengonfigurasi dan menghubungkan ESP32 ke platform Blynk. BLYNK\_TEMPLATE\_ID dan BLYNK\_TEMPLATE\_NAME mengidentifikasi template Blynk, sedangkan BLYNK\_AUTH\_TOKEN adalah token autentikasi yang unik. |
| #define BLYNK\_PRINT Serial | Code mengarahkan keluaran debug Blynk ke serial monitor untuk pemantauan. |
| #include <ESP32Servo.h>  #include <WiFi.h>  #include <BlynkSimpleEsp32.h> | #include <ESP32Servo.h>: Library untuk mengontrol servo dengan ESP32.  #include <WiFi.h>: Library untuk menghubungkan ESP32 ke jaringan WiFi.  #include <BlynkSimpleEsp32.h>: Library untuk menghubungkan ESP32 ke platform Blynk. |



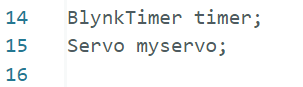
**Gambar 3.8 Variabel Yang Digunakan**

**Untuk Terhubung di Internet**

Berikut adalah penjelasan dari code / instruksi yang diberikan pada *gambar 3.8 :*

**Tabel 3.3 Variabel Yang Digunakan Untuk Terhubung di Internet**

|  |  |
| --- | --- |
| **Komponen** | **Keterangan** |
| char auth[] = BLYNK\_AUTH\_TOKEN;  char ssid[] = "IN";  char pass[] = "123456789"; | Baris code ini menjelaskan tentang bagaiamana perangakat terhubung dengan blynk dengan memasukan code token yang disediakan dari blynk dan menggunakan ssid serta password jaringan internet untuk menghubungkan blynk dengan perangkat |



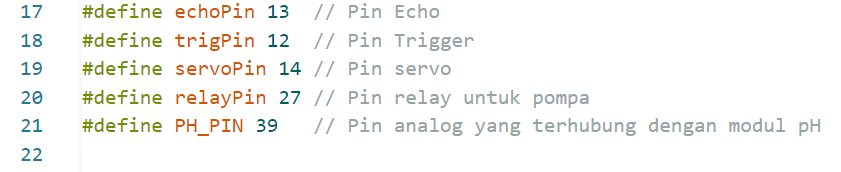
**Gambar 3.9 Pendeklarasian Variabel**

**Untuk Timer dan Motor Servo**

Berikut adalah penjelasan dari code / instruksi yang diberikan pada *gambar 3.9 :*

**Tabel 3.4 Pendeklarasian Variabel Untuk Timer dan Motor Servo**

|  |  |
| --- | --- |
| **Komponen** | **Keterangan** |
| BlynkTimer timer;  Servo myservo; | BlynkTimer dan Servo merupakan objeck yang digunakan untuk menkontrol : BlynkTimer timer, Objek timer untuk mengatur interval tugas (misalnya mengirim data sensor secara periodik).  Servo myservo, untuk menkontrol servo |



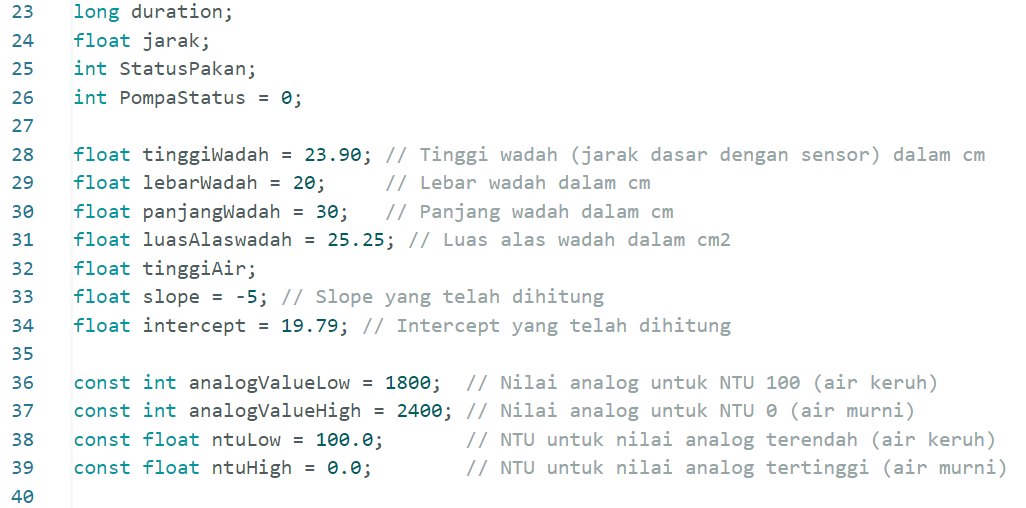
**Gambar 3.10 Pendefinisian Setiap Pin**

**Yang digunakan pada ESP32**

Berikut adalah penjelasan dari code / instruksi yang diberikan pada *gambar 3.10 :*

**Tabel 3.5 Pendefinisian Setiap Pin Yang digunakan pada ESP32**

|  |  |
| --- | --- |
| **Komponen** | **Keteranagn** |
| #define echoPin 13  // Pin Echo  #define trigPin 12  // Pin Trigger  #define servoPin 14 // Pin servo  #define relayPin 27 // Pin relay untuk pompa  #define PH\_PIN 39   // Pin analog yang terhubung dengan modul pH | Mendefinisikan pin pada komponen yang terhhubung dengan ESP32  Pin Echo Ultrasonik terhubung dengan pin D13 ESP32  Pin Trig Ultrasonik terhubung dengan pin D12 ESP32  Pin Sinyal pada servo terhubung dengan pin D14 ESP32  Pin IN pada relay terhubung dengan pin D27 ESP32  Pin Po pada modul pH terhubung dengan pin Vp (GPIO 39) pada ESP32 |



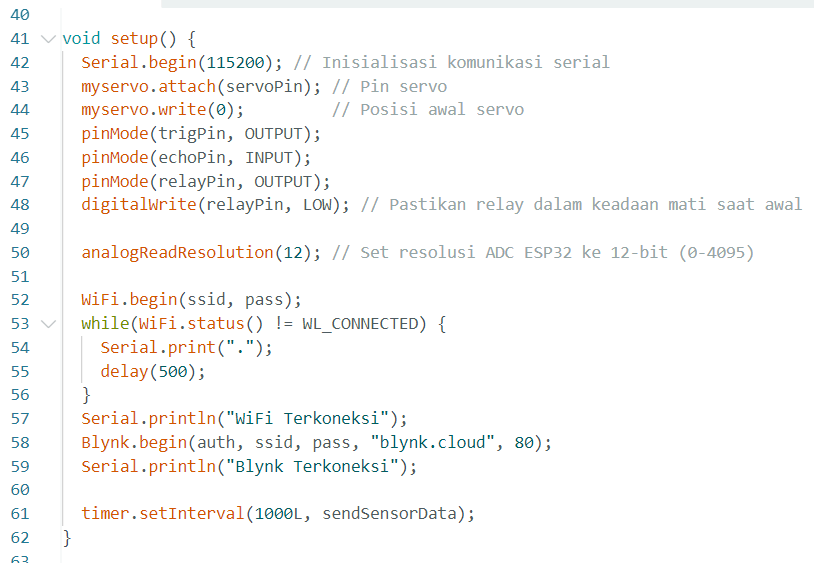
**Gambar 3.11 Mendeklarasikan Variabel pada**

**Setiap Objek yang Menyimpan Nilai Pengukuran**

Berikut adalah penjelasan dari code / instruksi yang diberikan pada *gambar 3.11 :*

**Tabel 3.6 Mendeklarasikan Variabel pada Setiap Objek yang Menyimpan Nilai Pengukuran**

|  |  |
| --- | --- |
| **Komponen** | **Keterangan** |
| long duration;  float jarak;  int StatusPakan;  int PompaStatus = 0; | long duration, float jarak merupkan variabel untuk menyimpan durasi sinyal ultrasonik dan jarak yang diukur.  Int StatusPakan menunjukan nilai variabel dari StatusPakan  int PompaStatus = 0 menunjukan nilai awal dari pompa itu 0 (aktif high) |
| float tinggiWadah = 23.90; // Tinggi wadah (jarak dasar dengan sensor) dalam cm  float lebarWadah = 20;     // Lebar wadah dalam cm  float panjangWadah = 30;   // Panjang wadah dalam cm  float luasAlaswadah = 25.25; // Luas alas wadah dalam cm2  float tinggiAir; | float tinggiWadah, float lebarWadah, float panjangWadah, float luasAlaswadah, float tinggiAir merupakan variabel untuk menyimpan parameter fisik wadah air. |
| float slope = -5; // Slope yang telah dihitung  float intercept = 19.79; // Intercept yang telah dihitung | float slope, float intercept merupakan parameter kalibrasi untuk sensor pH. |
| const int analogValueLow = 1800;  // Nilai analog untuk NTU 100 (air keruh)  const int analogValueHigh = 2400; // Nilai analog untuk NTU 0 (air murni)  const float ntuLow = 100.0;       // NTU untuk nilai analog terendah (air keruh)  const float ntuHigh = 0.0;        // NTU untuk nilai analog tertinggi (air murni) | const int analogValueLow, const int analogValueHigh, const float ntuLow, const float ntuHigh merupakan parameter kalibrasi untuk sensor kekeruhan. |

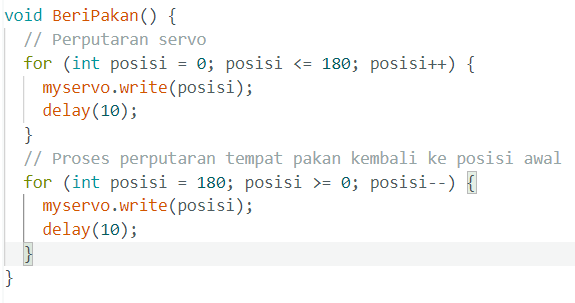


**Gambar 3.12 Perintah pada Void setup**

Berikut adalah penjelasan dari code / instruksi yang diberikan pada *gambar 3.12 :*

**Tabel 3.7 Perintah pada Void setup**

|  |  |
| --- | --- |
| **Code** | **Keterangan** |
| void setup() | void setup(), merupakan fungsi yang dijalankan sekali saat mikrokontroler dinyalakan atau di-reset. |
| Serial.begin(115200); | Serial.begin(115200), menginisialisasi komunikasi serial dengan baud rate 115200. |
| myservo.attach(servoPin);  myservo.write(0); | myservo.attach(servoPin), myservo.write(0), baris code ini menginisialisasi servo dan mengatur posisi awalnya ke 0 derajat. |
| pinMode(trigPin, OUTPUT);  pinMode(echoPin, INPUT);  pinMode(relayPin, OUTPUT); | pinMode, merupakan code untuk mengatur mode pin sebagai input atau output. |
| digitalWrite(relayPin, LOW); | digitalWrite(relayPin, LOW), code ini memastikan relay dalam keadaan mati saat awal (Low). |
| analogReadResolution(12); // Set resolusi ADC ESP32 ke 12-bit (0-4095) | analogReadResolution(12), mengatur resolusi ADC ESP32 ke 12-bit (0-4095). |
| WiFi.begin(ssid, pass); | WiFi.begin(ssid, pass), code untuk memulai koneksi WiFi dengan SSID dan password yang diberikan. |
| while(WiFi.status() != WL\_CONNECTED) {      Serial.print(".");      delay(500);    } | while(WiFi.status() != WL\_CONNECTED), code yang menunjukan proses ESP32 menunggu terhubung ke WiFi. |
| Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);    Serial.println("Blynk Terkoneksi"); | Blynk.begin(auth, ssid, pass, "blynk.cloud", 80): code untuk menghubungkan ESP32 ke Blynk. |
| timer.setInterval(1000L, sendSensorData); | timer.setInterval(1000L, sendSensorData), mengatur interval tugas untuk mengirim data sensor setiap 1 detik. |



**Gambar 3.13 Perintah pada Void BeriPakan untuk Mengkontrol**

**Pergerakan Motor Servo**

Berikut adalah penjelasan dari code / instruksi yang diberikan pada *gambar 3.13 :*

**Tabel 3.8 Perintah pada Void BeriPakan untuk Mengkontrol Pergerakan Motor Servo**

|  |  |
| --- | --- |
| **Code** | **Keterangan** |
| void BeriPakan() {    // Perputaran servo    for (int posisi = 0; posisi <= 180; posisi++) {      myservo.write(posisi);      delay(10);    }    // Proses perputaran tempat pakan kembali ke posisi awal    for (int posisi = 180; posisi >= 0; posisi--) {      myservo.write(posisi);      delay(10);    }  } | void BeriPakan(), code ini berfungsi untuk mengontrol pergerakan servo untuk memberikan pakan. Perulangan *for* digunakan untuk mencontrol perputaran servo dari posisi 0 derajat sampai posisi 180 derajat, kemudian kembali lagi dari 180 derajat ke 0 derajat |

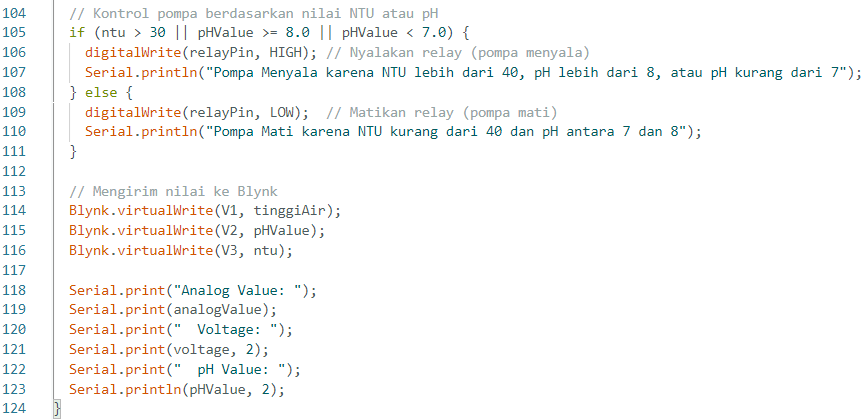


**Gambar 3.14 Perintah pada Void sendSensorData untuk Mengirim Data Semua Sensor Yang Digunakan**

Berikut adalah penjelasan dari code / instruksi yang diberikan pada *gambar 3.14 :*

**Tabel 3.9 Perintah pada Void sendSensorData untuk Mengirim Data Semua Sensor Yang Digunakan**

|  |  |
| --- | --- |
| **Code** | **Keterangan** |
| void sendSensorData() | Merupakan fungsi yang dijalankan secara periodik untuk mengukur dan mengirim data sensor ke Blynk. |
| // Mengukur jarak dengan sensor ultrasonik    digitalWrite(trigPin, LOW);    delayMicroseconds(2);    digitalWrite(trigPin, HIGH);    delayMicroseconds(10);    digitalWrite(trigPin, LOW);    duration = pulseIn(echoPin, HIGH);    jarak = duration \* 0.034 / 2; // Konversi ke jarak sebenarnya (cm) | Baris code ini menunjukan pengukuran nilai ketinggian air dengan menggunakan sensor ultrasonik, kemudian nilai yang dihasilkan dari pengukuran ultrasonik akan dikonversi dengan rumus yang sudah di definisikan pada variabel *jarak.* |
| // Menghitung tinggi air    tinggiAir = tinggiWadah - jarak; | Code ini digunakan setelah nilai konversi sensor ultrasonik diketahui, kemudian akan diukur ketinggian air pada aquariumnya |
| Blynk.run(); | Code untuk menjalankan Blynk |
| int sensorValue = analogRead(34);  // Membaca nilai analog dari pin 34 pada ESP32    // Kalibrasi nilai sensor ke NTU    float ntu = map(sensorValue, analogValueLow, analogValueHigh, ntuLow \* 100, ntuHigh \* 100) / 100.0;    Serial.print("TURBIDITY: ");    Serial.print(ntu);    Serial.println(" NTU"); | int sensorValue = analogRead(34), merupakan variabel untuk menghubungkan sensor Turbidity dengan ESP32 pada GPIO 34.  Fungsi *map* digunakan untuk menghitung nilai kekeruhan pada air aquarium |
| // Membaca nilai pH    int analogValue = analogRead(PH\_PIN); // Membaca nilai analog dari pin pH    float voltage = analogValue \* (3.3 / 4095.0); // Konversi nilai ADC ke tegangan    float pHValue = slope \* voltage + intercept; // Menghitung nilai pH berdasarkan tegangan | Code untuk membaca nilai pH air aquarium |



**Gambar 3.15 Kondisi Untuk Menyalakan Pompa Aquarium dan Mengirim**

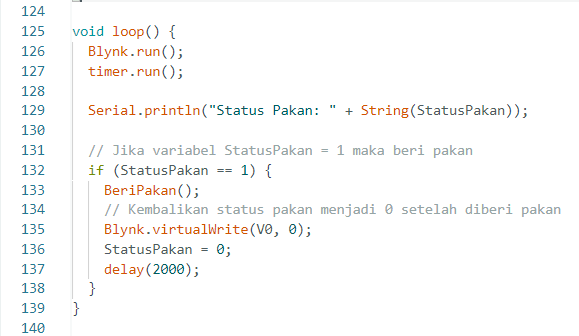
**Nilai ke Blynk**

Berikut adalah penjelasan dari code / instruksi yang diberikan pada *gambar 3.15 :*

**Tabel 3.10 Kondisi Untuk Menyalakan Pompa Aquarium dan Mengirim**

**Nilai ke Blynk**

|  |  |
| --- | --- |
| **Code** | **Keterangan** |
| // Kontrol pompa berdasarkan nilai NTU atau pH    if (ntu > 30 || pHValue >= 8.0 || pHValue < 7.0) {      digitalWrite(relayPin, HIGH); // Nyalakan relay (pompa menyala)      Serial.println("Pompa Menyala karena NTU lebih dari 40, pH lebih dari 8, atau pH kurang dari 7");    } else {      digitalWrite(relayPin, LOW);  // Matikan relay (pompa mati)      Serial.println("Pompa Mati karena NTU kurang dari 40 dan pH antara 7 dan 8");    } | Mengkontrol pompa sesuai dengan kondisi yang diinginkan |
| // Mengirim nilai ke Blynk    Blynk.virtualWrite(V1, tinggiAir);    Blynk.virtualWrite(V2, pHValue);    Blynk.virtualWrite(V3, ntu); | Mengirim data ke blynk melalui pin virtual yang disediakan oleh blynk (V1, V2 dan V3). |
| Serial.print("Analog Value: ");    Serial.print(analogValue);    Serial.print("  Voltage: ");    Serial.print(voltage, 2);    Serial.print("  pH Value: ");    Serial.println(pHValue, 2); | Menampilkan nilai output komponen pada serial monitor |



**Gambar 3.16 Perintah Pada void loop untuk menjalankan Blynk dan untuk**

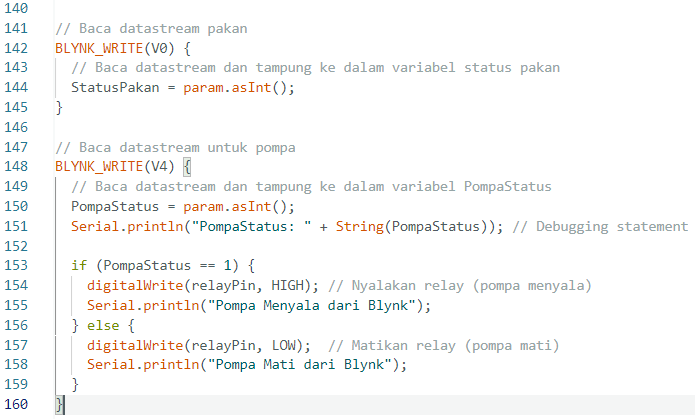
**Mengkontrol Motor Servo**

Berikut adalah penjelasan dari code / instruksi yang diberikan pada *gambar 3.16 :*

**Tabel 3.11 Perintah Pada void loop untuk menjalankan Blynk dan untuk**

**Mengkontrol Motor Servo**

|  |  |
| --- | --- |
| **Code** | **Keterangan** |
| void loop() {    Blynk.run();    timer.run();    Serial.println("Status Pakan: " + String(StatusPakan));      // Jika variabel StatusPakan = 1 maka beri pakan    if (StatusPakan == 1) {      BeriPakan();      // Kembalikan status pakan menjadi 0 setelah diberi pakan      Blynk.virtualWrite(V0, 0);      StatusPakan = 0;      delay(2000);    }  } | Memastikan Blynk dan timer terus berjalan. Jika StatusPakan adalah 1, maka fungsi BeriPakan dipanggil dan StatusPakan di-reset ke 0. |



**Gambar 3.17 Membaca datastream Pada Pin V0 (Motor Servo)**

**Dan Pin V4 (Relay)**

Berikut adalah penjelasan dari code / instruksi yang diberikan pada *gambar 3.17 :*

**Tabel 3.12 Membaca datastream Pada Pin V0 (Motor Servo)**

**Dan Pin V4 (Relay)**

|  |  |
| --- | --- |
| **Code** | **Keterangan** |
| // Baca datastream pakan  BLYNK\_WRITE(V0) {    // Baca datastream dan tampung ke dalam variabel status pakan    StatusPakan = param.asInt();  } | Membaca void StatusPakan pada pin virtual V0 |
| // Baca datastream untuk pompa  BLYNK\_WRITE(V4) {    // Baca datastream dan tampung ke dalam variabel PompaStatus    PompaStatus = param.asInt();    Serial.println("PompaStatus: " + String(PompaStatus)); // Debugging statement | Membaca PompaStatus ke blynk melalui pin virtual V4 |
| if (PompaStatus == 1) {      digitalWrite(relayPin, HIGH); // Nyalakan relay (pompa menyala)      Serial.println("Pompa Menyala dari Blynk");    } else {      digitalWrite(relayPin, LOW);  // Matikan relay (pompa mati)      Serial.println("Pompa Mati dari Blynk");    }  } | Kondisi untuk menjalankan pompa, jika relay berlogika 1 maka pompa akan menyala |

**3.3 Pengujian Alat**