

Perancangan dan Implementasi Sistem Deteksi Ketinggian Air Berbasis Sensor Ultrasonik

oleh

Maynanda Elisa Pasya¹ Nur Husnil Khotimah Harahap² Nadia Alya Paramitha Erwanto³

Fakultas Vokasi, Universitas Brawijaya

Email: nadiaalya2729@student.ub.ac.id, husnilharahap@student.ub.ac.id, maynandatbg@student.ub.ac.id

Abstrak

Proyek ini bertujuan untuk merancang dan mengimplementasikan sistem monitoring jarak permukaan air berbasis Internet of Things (IoT) dengan menggunakan sensor ultrasonik. Sistem ini dirancang untuk memantau ketinggian air secara real-time dan mengirimkan data hasil pengukuran ke aplikasi Blynk melalui koneksi internet. Informasi yang ditampilkan pada aplikasi dapat membantu pengguna untuk mengetahui secara langsung kapan wadah air perlu diisi ulang atau kapan wadah sudah penuh, sehingga dapat mencegah terjadinya tumpahan atau kekurangan air yang dapat mengganggu aktivitas sehari-hari. Sistem ini menggunakan platform pengembangan Visual Studio Code dengan simulasi berbasis Wokwi, sehingga memungkinkan proses pengujian dilakukan secara virtual sebelum diterapkan pada perangkat fisik. Hasil implementasi menunjukkan bahwa sistem mampu mendeteksi perubahan ketinggian air secara akurat dan mengirimkan data dengan respon cepat dan stabil. Sistem ini memiliki potensi untuk dikembangkan lebih lanjut dalam skala nyata, dengan dukungan sensor dan mikrokontroler fisik yang dapat digunakan pada berbagai jenis wadah air seperti tangki, toren, atau tempat penampungan lainnya untuk keperluan rumah tangga maupun industri.

Kata Kunci: IoT, monitoring ketinggian air, jarak air, sensor ultrasonik, Blynk, Wokwi

Abstract

This project aims to design and implement a water level distance monitoring system based on the Internet of Things (IoT) using an ultrasonic sensor. The system is designed to monitor water height in real-time and transmit measurement data to the Blynk application via an internet connection. The information displayed helps users determine when the water container needs refilling or when it is already full, thereby preventing overflow or water shortages that could disrupt daily activities. The development platform used is Visual Studio Code with Wokwi-based simulation, allowing virtual testing before physical deployment. The implementation results show that the system can accurately detect water level changes and transmit data with fast and stable response. This system has the potential to be further developed for real-world applications using physical sensors and microcontrollers that can be applied to various types of water containers, such as tanks, reservoirs, or other storage units for household or industrial purposes.

Keywords: IoT, water level monitoring, water distance, ultrasonic sensor, Blynk, Wokwi

Pendahuluan

Menjaga kebutuhan dasar hewan peliharaan adalah bagian penting dari tanggung jawab seorang pemilik, termasuk memastikan hewan seperti kucing dan anjing mendapatkan asupan cairan yang cukup setiap harinya. Air minum yang bersih berperan penting dalam menjaga fungsi organ tubuh hewan, mulai dari kesehatan ginjal hingga sistem pencernaan dan metabolisme. Jika kebutuhan tersebut tidak terpenuhi, hewan dapat mengalami dehidrasi yang berpotensi membahayakan kesehatannya..

Namun, dalam kehidupan sehari-hari, pemilik hewan sering kali memiliki aktivitas yang padat atau bepekerjaan, sehingga tidak selalu dapat memantau ketersediaan air minum di wadah hewan secara rutin. Hal ini dapat menyebabkan air dalam wadah habis tanpa disadari, yang tentu dapat membahayakan kondisi hewan peliharaan. Selain itu, ada pula risiko air tumpah atau terkontaminasi jika tidak dikelola dengan baik.

Selama ini, pengisian air untuk hewan peliharaan masih banyak dilakukan secara manual. Cara ini cenderung kurang efisien karena memerlukan perhatian dan keterlibatan langsung dari pemilik. Ketika pemilik lengah atau tidak berada di rumah, hewan bisa saja kehausan karena wadah air kosong, atau sebaliknya, air terus mengalir tanpa kontrol dan menyebabkan pemborosan.

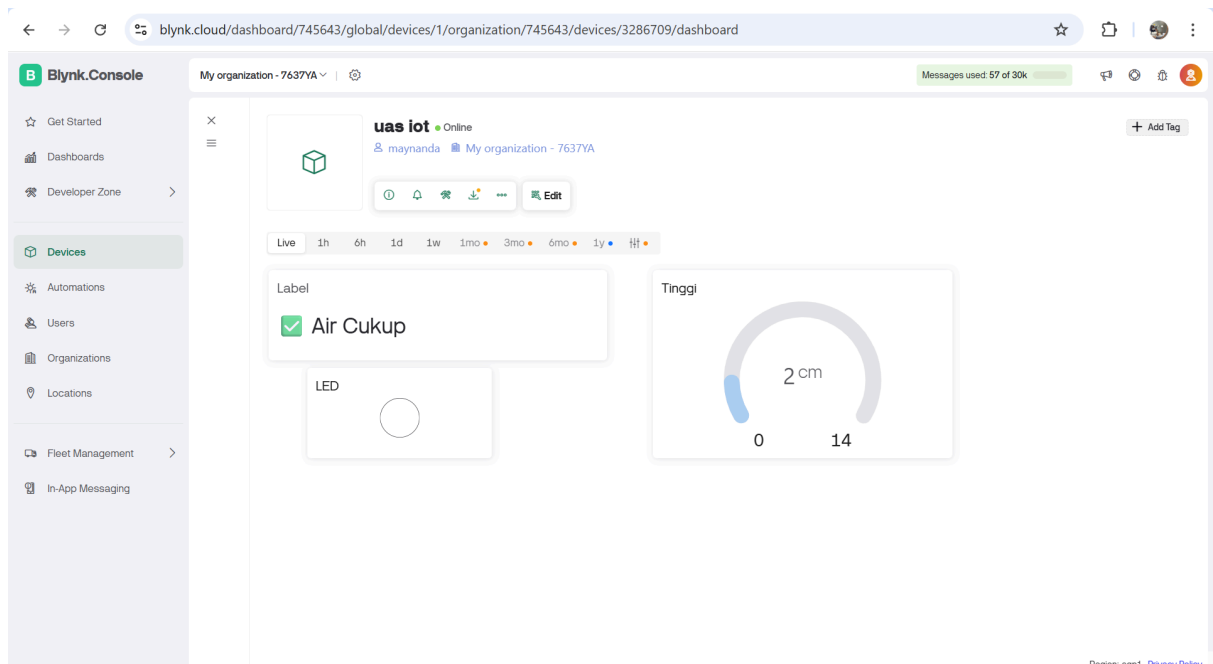
Dengan berkembangnya teknologi Internet of Things (IoT), muncul peluang untuk merancang sistem otomatis yang dapat membantu pemilik hewan dalam memantau dan mengelola kebutuhan air peliharaan mereka secara real-time dan jarak jauh. Sistem berbasis IoT dapat mengintegrasikan sensor, mikrokontroler, serta koneksi internet untuk mengukur level air secara otomatis dan mengirimkan notifikasi kepada pengguna melalui aplikasi mobile.

Berdasarkan permasalahan tersebut, proyek ini bertujuan untuk merancang dan mengimplementasikan sistem pengisian dan monitoring air minum otomatis untuk hewan peliharaan berbasis IoT. Sistem ini menggunakan mikrokontroler ESP32 yang terhubung ke internet, serta memanfaatkan sensor ultrasonik untuk mengukur ketinggian air di dalam wadah. Data pengukuran dikirimkan secara real-time ke aplikasi Blynk sebagai antarmuka pemantauan. Dengan adanya sistem ini, pemilik hewan dapat lebih tenang karena kebutuhan air peliharaan dapat terpantau secara otomatis, lebih efisien, serta mengurangi risiko kehabisan atau pemborosan air. Sistem ini juga dapat menjadi solusi yang praktis, terutama bagi mereka yang memiliki mobilitas tinggi atau memelihara lebih dari satu hewan dalam satu waktu.

Metodologi

Metodologi yang digunakan dalam perancangan sistem monitoring ketinggian air ini melibatkan integrasi antara perangkat keras dan perangkat lunak berbasis IoT. Sistem terdiri dari mikrokontroler ESP32, sensor ultrasonik HC-SR04, LED indikator, dan aplikasi Blynk sebagai antarmuka pemantauan. Sensor HC-SR04 bekerja dengan memancarkan gelombang ultrasonik yang dipantulkan kembali saat mengenai permukaan air; ESP32 kemudian menghitung waktu tempuh gelombang untuk menentukan jarak air, mengonversinya menjadi data ketinggian, dan mengirimkannya ke aplikasi Blynk melalui koneksi Wi-Fi. Data ditampilkan secara real-time di dashboard aplikasi, sementara LED indikator menyala jika ketinggian air berada dalam batas tertentu (misalnya terlalu rendah atau hampir penuh) sebagai peringatan visual. Perangkat lunak dikembangkan menggunakan PlatformIO di Visual Studio Code, dengan pustaka seperti WiFi.h, BlynkSimpleEsp32.h, dan NewPing.h untuk mendukung fungsi konektivitas dan pembacaan sensor. Sebelum implementasi fisik, simulasi sistem dilakukan menggunakan Wokwi untuk memastikan logika berjalan sesuai yang diharapkan. Output dari sistem ini berupa tampilan data ketinggian air secara real-time di aplikasi, indikator LED, serta opsi notifikasi jika fitur tersebut diaktifkan.

Hasil dan Pembahasan



The screenshot shows the Arduino IDE interface with the code for the "uas iot" device. The code is written in C++ and includes the following components:

```
1 #define BLYNK_TEMPLATE_ID "TMPL6VnDP64ZE"
2 #define BLYNK_TEMPLATE_NAME "uas iot"
3 #define BLYNK_AUTH_TOKEN "LQ8DzxhiHyVvPPTuAky24kW0m24fOFdO"
4
5 #include <WiFi.h>
6 #include <BlynkSimpleEsp32.h>
7
8 char auth[] = BLYNK_AUTH_TOKEN;
9 char ssid[] = "ur area";
10 char pass[] = "ripka123";
11
12 #define TRIG_PIN 13
13 #define ECHO_PIN 27
14
15 #define LED_AIR_HABIS 25 // LED indikator air habis (pakai pin 25)
16
17 void setup() {
18   Serial.begin(115200);
19   delay(1000); // Tambahan delay untuk memastikan serial aktif
20 }
```

The output window shows the following messages:

```
Status air: [X] Air Cukup
Jarak air: 0.00 cm
Status air: [X] Air Cukup
Jarak air: 0.00 cm
Status air: [X] Air Cukup
Jarak air: 0.00 cm
Status air: [X] Air Cukup
Jarak air: 0.00 cm
Status air: [X] Air Cukup
Jarak air: 0.00 cm
```

The bottom status bar indicates the code is on line 73, column 1, and the device is connected to the ESP32 Dev Module on COM3 at 115200 baud.

uas | Arduino IDE 2.3.6

File Edit Sketch Tools Help

ESP32 Dev Module

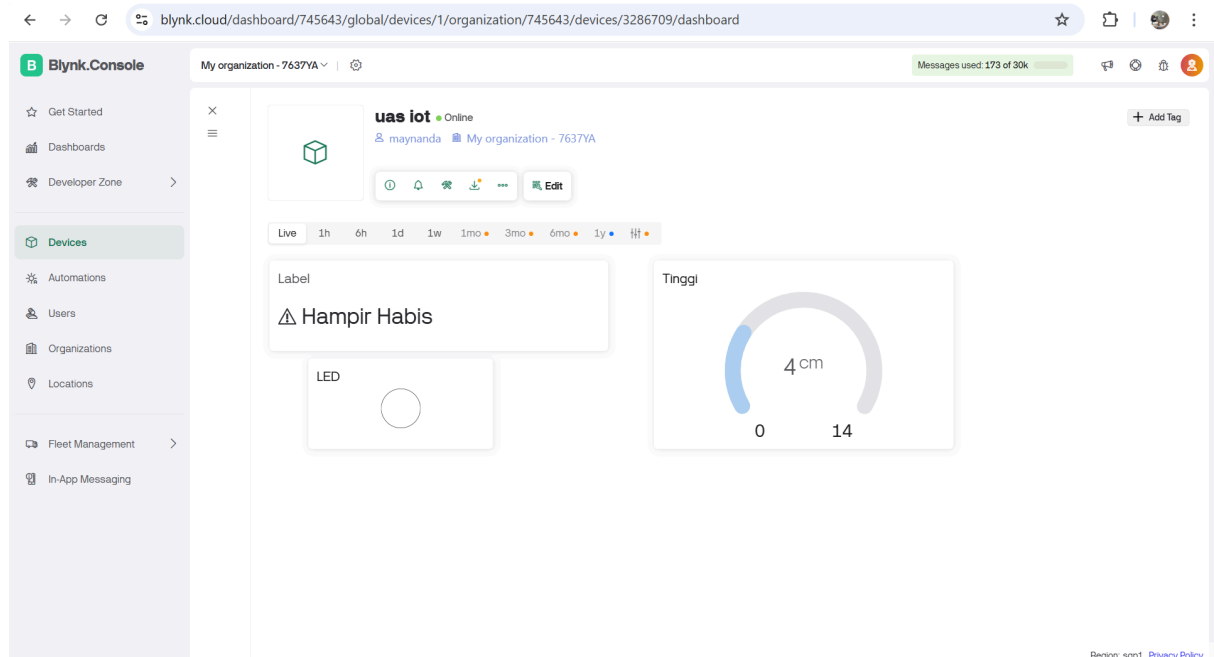
```
1 #define BLYNK_TEMPLATE_ID "TMPL6VnDP64ZE"
2 #define BLYNK_TEMPLATE_NAME "uas iot"
3 #define BLYNK_AUTH_TOKEN "LQ8DzxhiHyVvPPTuAky24Ww8m24f0F0d0"
4
5 #include <WiFi.h>
6 #include <BlynkSimpleEsp32.h>
7
8 char auth[] = BLYNK_AUTH_TOKEN;
9 char ssid[] = "ur_area";
10 char pass[] = "ripka123";
11
12 #define TRIG_PIN 13
13 #define ECHO_PIN 27
14
15 #define LED_AIR_HABIS 25 // LED indikator air habis (pakai pin 25)
16
17 void setup() {
18   Serial.begin(115200);
19   delay(1000); // Tambahan delay untuk memastikan serial aktif
20 }
```

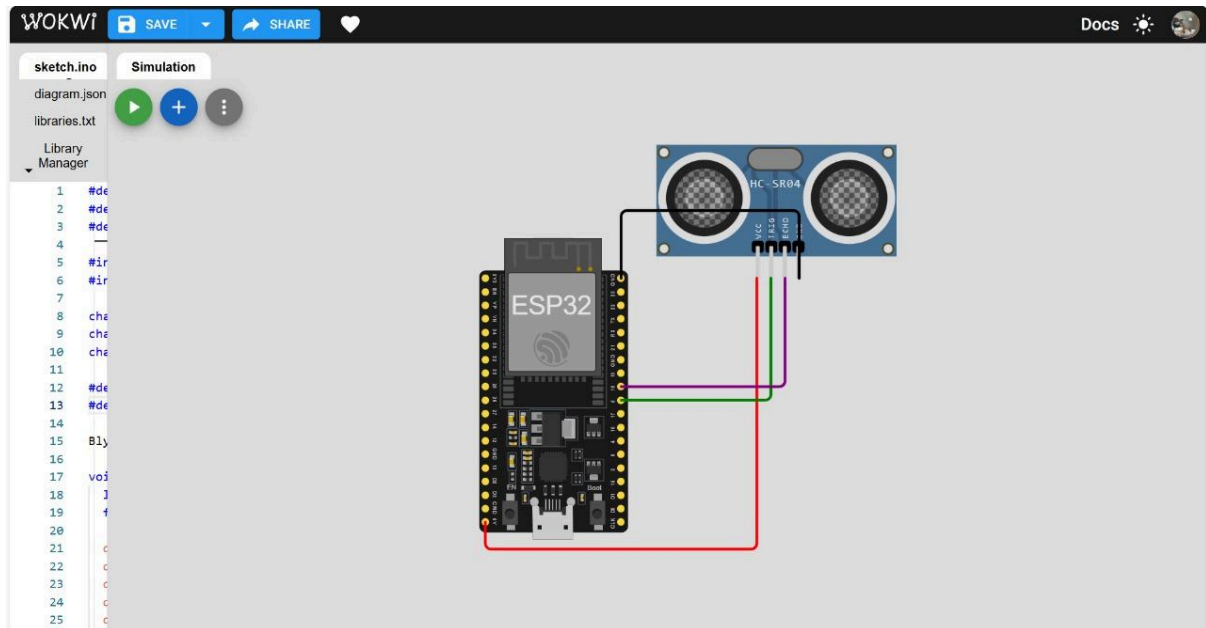
Output Serial Monitor X

Message (Enter to send message to 'ESP32 Dev Module' on 'COM3')

Status air: Δ Hampir Habis
Jarak air: 5.54 cm
Status air: Δ Hampir Habis
Jarak air: 6.51 cm
Status air: ▲ Isi Air!
Jarak air: 5.54 cm
Status air: Δ Hampir Habis
Jarak air: 5.54 cm
Status air: Δ Hampir Habis

In 16, Col 1 - ESP32 Dev Module on COM3





Proyek ini bertujuan mendeteksi ketinggian air minum hewan menggunakan sensor HC-SR04 dan menampilkan data ke aplikasi Blynk. Pada implementasi awal, muncul kendala seperti hasil jarak “0 cm”, error timeout, dan data tidak masuk ke Blynk.

Setelah dilakukan pengujian dan perbaikan, dilakukan beberapa perubahan penting: penambahan LED dan resistor sebagai indikator dan penstabil tegangan, penggantian pin TRIG dan ECHO ke pin yang lebih stabil, penambahan pembagi tegangan untuk mengamankan input ESP32, perubahan suplai sensor dari 3.3V ke 5V, serta perbaikan jalur GND. Hasilnya, sistem menjadi lebih stabil, akurat, dan sesuai tujuan awal.

No.	Jenis Uji Coba	Langkah Uji	Hasil yang Diharapkan
1	Kondisi Awal	Sistem dinyalakan dan terhubung ke WiFi	ESP32 mendapatkan IP, Serial Monitor menampilkan "WiFi Terhubung <input checked="" type="checkbox"/> " dan "Mulai membaca jarak air..."
2	Pengujian Sensor Jarak	Gelas/mangkuk berisi air diletakkan di bawah sensor	Sensor membaca jarak, LED indikator menyala sesuai ambang batas jarak
3	Pengujian Koneksi Blynk	Perubahan level air diamati di dashboard Blynk (Virtual Pin V0 dan V1)	Data tampil real-time dan berubah sesuai level air tanpa delay atau disconnect

4	Pengujian Ketahanan	Sistem dibiarkan menyala selama lebih dari 10 menit	Sistem tetap stabil, tidak overheat, tidak reboot otomatis
5	Pengujian Kegagalan	Air dijauhkan atau sensor dilepas	Muncul status "🚰 Isi Air!" saat air jauh

Seluruh fungsi berjalan baik, komunikasi ke Blynk stabil, dan pengguna dapat memantau kondisi air secara real-time melalui ponsel. Sistem berhasil memenuhi tujuan awal dengan hasil yang aman, efisien, dan handal.

Lampiran

Kode Program :

```
#define BLYNK_TEMPLATE_ID "TMPL6VnDP64ZE"

#define BLYNK_TEMPLATE_NAME "uas iot"

#define BLYNK_AUTH_TOKEN "LQ8DzxhiHyVvPPTuAky24WW0m24fOFdO"

#include <WiFi.h>

#include <BlynkSimpleEsp32.h>

char auth[] = BLYNK_AUTH_TOKEN;

char ssid[] = "ur area";

char pass[] = "ripka123";

#define TRIG_PIN 13

#define ECHO_PIN 27

#define LED_AIR_HABIS 25 // LED indikator air habis (pakai pin 25)
```

```
void setup() {  
  
    Serial.begin(115200);  
  
    delay(1000); // Tambahan delay untuk memastikan serial aktif  
  
    Serial.println("Memulai program...");  
  
  
    pinMode(TRIG_PIN, OUTPUT);  
  
    pinMode(ECHO_PIN, INPUT);  
  
    pinMode(LED_AIR_HABIS, OUTPUT);  
  
    digitalWrite(LED_AIR_HABIS, LOW); // Pastikan LED mati di awal  
  
  
    Blynk.begin(auth, ssid, pass);  
  
  
    Serial.println("Terhubung ke WiFi dan Blynk.");  
  
    Serial.println("Mulai membaca jarak air...");  
  
}  
  
  
void loop() {  
  
    Blynk.run();  
  
  
    long duration;  
  
    float distance;
```



```
digitalWrite(TRIG_PIN, LOW);

delayMicroseconds(2);

digitalWrite(TRIG_PIN, HIGH);

delayMicroseconds(10);

digitalWrite(TRIG_PIN, LOW);


duration = pulseIn(ECHO_PIN, HIGH);

distance = duration * 0.034 / 2;


Serial.print("Jarak air: ");

Serial.print(distance);

Serial.println(" cm");


String statusAir;

if (distance < 3) {

    statusAir = "✅ Air Cukup";

    digitalWrite(LED_AIR_HABIS, LOW); // LED mati

} else if (distance <= 6) {

    statusAir = "⚠ Hampir Habis";

    digitalWrite(LED_AIR_HABIS, LOW); // LED mati

} else {

    statusAir = "💧 Isi Air!";

    digitalWrite(LED_AIR_HABIS, HIGH); // LED menyala

}
```

```
Serial.print("Status air: ");
```

```
Serial.println(statusAir);
```

```
Blynk.virtualWrite(V0, distance);
```

```
Blynk.virtualWrite(V1, statusAir);
```

```
delay(2000);
```

```
}
```

Dokumentasi pengujian :

