

# **Realtime Flood Monitoring System**

## **PHP, MySQLi, NodeMCU / ESP32 & Ultrasonic Sensor**

## Real time Flood Monitoring Database Example

### PHP, MySQLi, NodeMCU/ESP32 & Ultrasonic Sensor (HC-SR04)

#### SCENARIO

This is a tutorial on how to setup a real time flood database system comprising of HC-SR04 sensor x 1, NodeMCU x1/ ESP32 x1, PHP & MySQLi.

#### A. Setting Up Database

Table 1: Database properties and setup.

Database Name	river_level_data		
Table Name	level_status_daerah1		
<b>5 columns</b>			
NAME	TYPE	LENGTH	ADDITIONAL SETTING
dataID	INT	11	Index: <b>PRIMARY</b> A.I.: ✓
log	DATETIME	6	Default: <b>CURRENT_TIMESTAMP</b> Attributes: <b>on update CURRENT_TIMESTAMP</b>
river_name	VARCHAR	100	
water_level	VARCHAR	100	
river_status	VARCHAR	100	

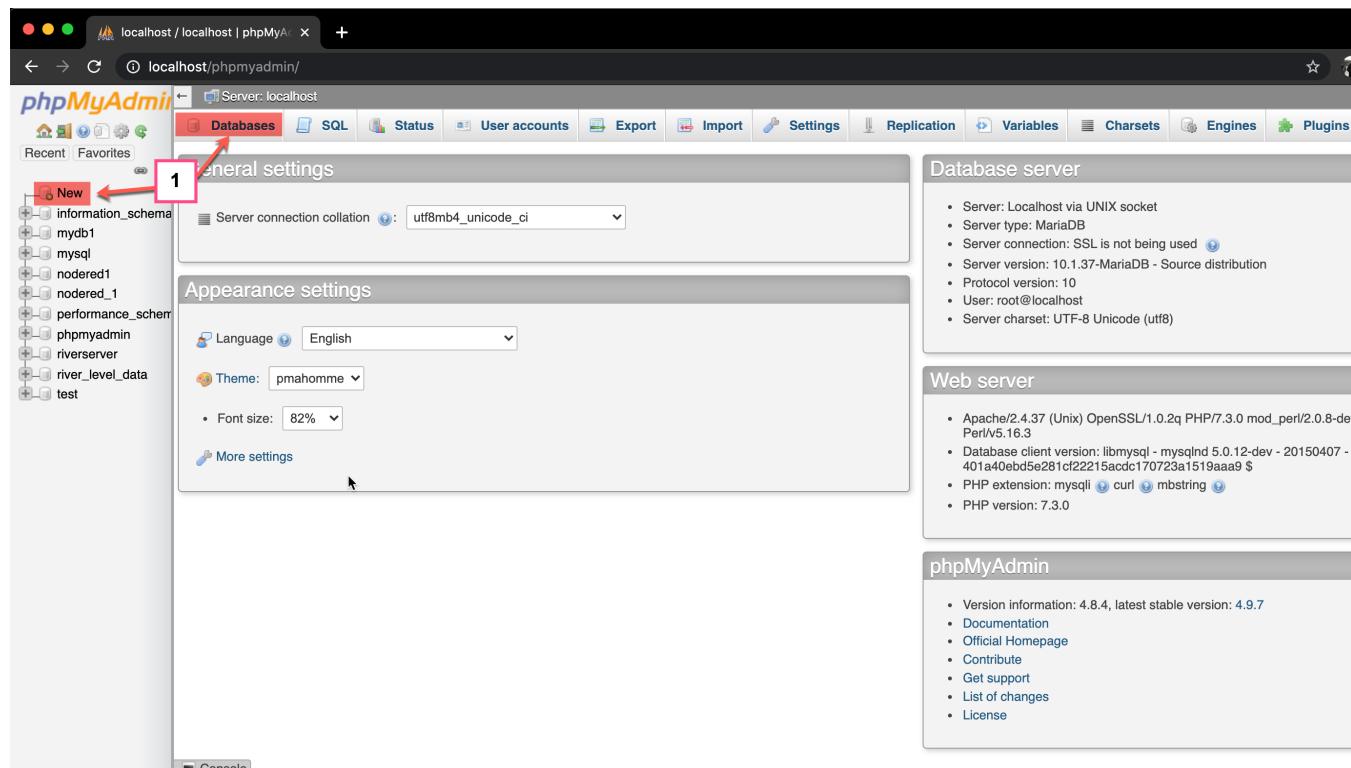
\*\*A.I. = Auto Increment

#### Step 1:

Start Apache Web Server & MySQL Database service.

Open your favourite browser, type **localhost/phpMyAdmin** at the address bar and press ENTER.

Click either **New** or **Databases** link / tab.



The screenshot shows the phpMyAdmin interface with the following details:

- Left sidebar:** Shows a tree view of databases: information\_schema, mydb1, mysql, nodered1, nodered\_1, performance\_schema, phpmyadmin, riverserver, river\_level\_data, and test.
- Main navigation:** Shows tabs for Databases, SQL, Status, User accounts, Export, Import, Settings, Replication, Variables,Charsets, Engines, and Plugins.
- Current page:** Databases
- Buttons:** New (highlighted with a red box and number 1), Import, Export, Settings, Replication, Variables,Charsets, Engines, and Plugins.
- General settings:** Server connection collation is set to utf8mb4\_unicode\_ci.
- Appearance settings:** Language is English, Theme is pmahomme, and Font size is 82%.
- Database server:** Localhost via UNIX socket, MariaDB, SSL not being used, version 10.37-MariaDB, User root@localhost, Charset UTF-8 Unicode (utf8).
- Web server:** Apache/2.4.37 (Unix) OpenSSL/1.0.2q PHP/7.3.0 mod\_perl/2.0.8-dev Perl/v5.16.3.
- phpMyAdmin:** Version information: 4.8.4, latest stable version: 4.9.7.

## Real time Flood Monitoring Database Example

### PHP, MySQLi, NodeMCU/ESP32 & Ultrasonic Sensor (HC-SR04)

#### Step 2:

Database name: **river\_level\_data**

The screenshot shows the phpMyAdmin interface on a Mac OS X desktop. The title bar says "localhost / localhost | phpMyAdmin". The main menu has "Databases" selected. On the left, a tree view shows various databases like information\_schema, mydb1, mysql, nodered1, nodered\_1, performance\_schema, phpmyadmin, riverserver, and test. In the center, there's a "Create database" form with "river\_level\_data" in the input field (marked with a yellow box labeled 2) and "latin1\_swedish\_ci" in the dropdown (marked with a pink box labeled 3). Below the form is a table of existing databases with their collations and actions. At the bottom, there are checkboxes for "Check all" and "With selected:" followed by "Drop" and "Console" buttons.

#### Step 3:

Table name: **level\_status\_daerah1**

Number of columns: **5**

The screenshot shows the phpMyAdmin interface on a Mac OS X desktop. The title bar says "Server: localhost > Database: river\_level\_data". The main menu has "Structure" selected. On the left, the same tree view of databases is shown. In the center, there's a "Create table" form with "Name: level\_status\_daerah1" (marked with a yellow box labeled 4) and "Number of columns: 5" (marked with a yellow box labeled 5). Below the form is a large empty table structure area. At the bottom right, there is a "Go" button (marked with a pink box labeled 6).

## Real time Flood Monitoring Database Example

### PHP, MySQLi, NodeMCU/ESP32 & Ultrasonic Sensor (HC-SR04)

#### Step 4:

Fill up the fields.

A pop up window will appear after item #10. Click **OK** (#11).

Leave #15 BLANK.

#### Step 5:

The database for the system has been created and currently under **Structure** tab.

Click **Browse** tab (#28) for next step.

## Real time Flood Monitoring Database Example

### PHP, MySQLi, NodeMCU/ESP32 & Ultrasonic Sensor (HC-SR04)

#### Step 6:

The database still empty since no data added into it.

Click **Insert** tab (#29) for manually key in the data.

The screenshot shows the phpMyAdmin interface for a MySQL database named 'river\_level\_data'. The 'level\_status\_daerah1' table is selected. The 'Insert' tab is highlighted with a red arrow and the number '29'. A yellow box highlights the table structure with the text 'Empty database.' A yellow arrow points from the text to the table header. The table has columns: dataID, log, river\_name, water\_level, and river\_status. The dataID column is highlighted in red. The log, river\_name, water\_level, and river\_status columns are in black. The table body is empty. The left sidebar shows various databases and tables. The bottom right corner has a watermark: 'Bookmark this SQL query'.

## Real time Flood Monitoring Database Example

PHP, MySQLi, NodeMCU/ESP32 & Ultrasonic Sensor (HC-SR04)

### Step 7:

Key in the data as #30 - #35. This step is important since it can also test the reliability of the database.

Column	Type	Function	Null	Value
dataID	int(11)		Null	
log	datetime(6)		Null	CURRENT_TIMESTAMP(6.000)
river_name	varchar(100)		Null	river #1
water_level	varchar(100)		Null	level #1
river_status	varchar(100)		Null	status #1

Do not put any value.  
An Auto Increment number for data ID.

Column	Type	Function	Null	Value
dataID	int(11)		Null	
log	datetime(6)		Null	CURRENT_TIMESTAMP(6.000)
river_name	varchar(100)		Null	river #2
water_level	varchar(100)		Null	level #2
river_status	varchar(100)		Null	status #2

Do not put any value.  
Time & Date save into the database  
- will follow pc/laptop setting.

30      31      32

33      34      35

36      Go

## Real time Flood Monitoring Database Example

### PHP, MySQLi, NodeMCU/ESP32 & Ultrasonic Sensor (HC-SR04)

#### Step 8:

The information in Step 7 have successfully inserted into **table level\_status\_daerah1**.

The screenshot shows the phpMyAdmin interface for the 'river\_level\_data' database. The 'level\_status\_daerah1' table is selected. The status bar at the top indicates '2 rows inserted.' A yellow box highlights this message, and another yellow box highlights the SQL syntax below it. The SQL query shown is:

```
INSERT INTO `level_status_daerah1` (`dataID`, `log`, `river_name`, `water_level`, `river_status`) VALUES (NULL, CURRENT_TIMESTAMP, 'river #1', 'level #1', 'status #1'), (NULL, CURRENT_TIMESTAMP, 'river #2', 'level #2', 'status #2');
```

The results pane on the right shows the columns: dataID, log, river\_name, water\_level, and river\_status. The table structure is also visible.

#### Step 9:

You may click the **Insert** tab if you wish to add another data manually.

The screenshot shows the phpMyAdmin interface for the 'river\_level\_data' database. The 'level\_status\_daerah1' table is selected. The status bar at the top indicates 'Showing rows 0 - 1 (2 total, Query took 0.0005 seconds.)'. A yellow box highlights this message. The table data is:

	dataID	log	river_name	water_level	river_status
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	1	2020-11-05 14:06:51	river #1	level #1	status #1
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	2	2020-11-05 14:06:51	river #2	level #2	status #2

A yellow arrow points from the bottom of the previous screenshot to the 'Edit' link for the second row. The bottom section of the screenshot shows the 'Query results operations' and 'Bookmark this SQL query' sections.

## Real time Flood Monitoring Database Example

### PHP, MySQLi, NodeMCU/ESP32 & Ultrasonic Sensor (HC-SR04)

Step 10:

**REMINDED:**  
EMPTY will erase all data in the respective table and set the Auto Increment to 1.  
DROP will delete the respective table

**END OF PART A.**

## B. PHP Files

Step 1:

Download Sublime for editor to PHP files.

Step 2:

Find where XAMPP was install. Normally at C folder.

Step 3:

Find **htdocs** folder. **C://xampp/htdocs**

Step 4:

Create a working folder inside the htdocs. You are going to save all your works inside this working folder. Create new working folder for new projects. Do not mix up new projects with the existing projects.

**C://xampp/htdocs/a\_flood/**

Step 5:

Save **inc.inc**, **add.php** and **get\_status.php** inside the **a\_flood** folder.

You can use Notepad, Notepad++, Sublime, Atom, Visual Studio Code to edit the PHP files.

Table 2: PHP files and its description

<b>Filename</b>	<b>Description</b>
inc.inc	Database configuration / setup
add.php	As medium to add data from sensor (Nodemcu) into server
get_status.php	Nodemcu request status from server

## Real time Flood Monitoring Database Example

### PHP, MySQLi, NodeMCU/ESP32 & Ultrasonic Sensor (HC-SR04)

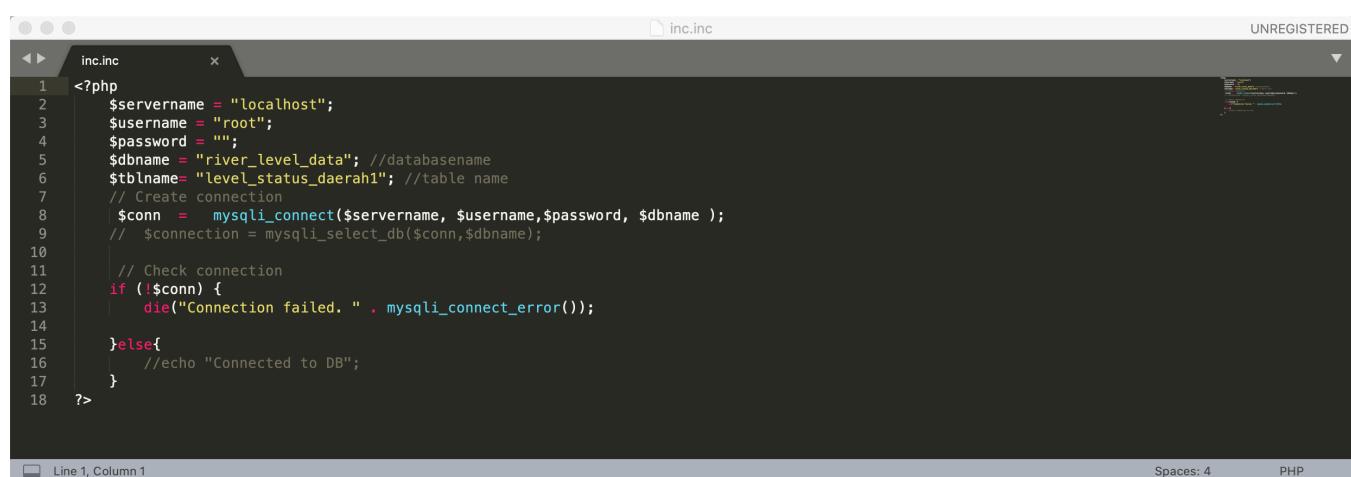
Step 7:

Filename: inc.inc ← save inside c://xampp/htdocs/a\_flood/

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "river_level_data"; //databasename
$tblname= "level_status_daerah1"; //table name
// Create connection
$conn = mysqli_connect($servername, $username,$password, $dbname );
// $connection = mysqli_select_db($conn,$dbname);

// Check connection
if (!$conn) {
die("Connection failed. " . mysqli_connect_error());

} else{
//echo "Connected to DB";
}
?>
```

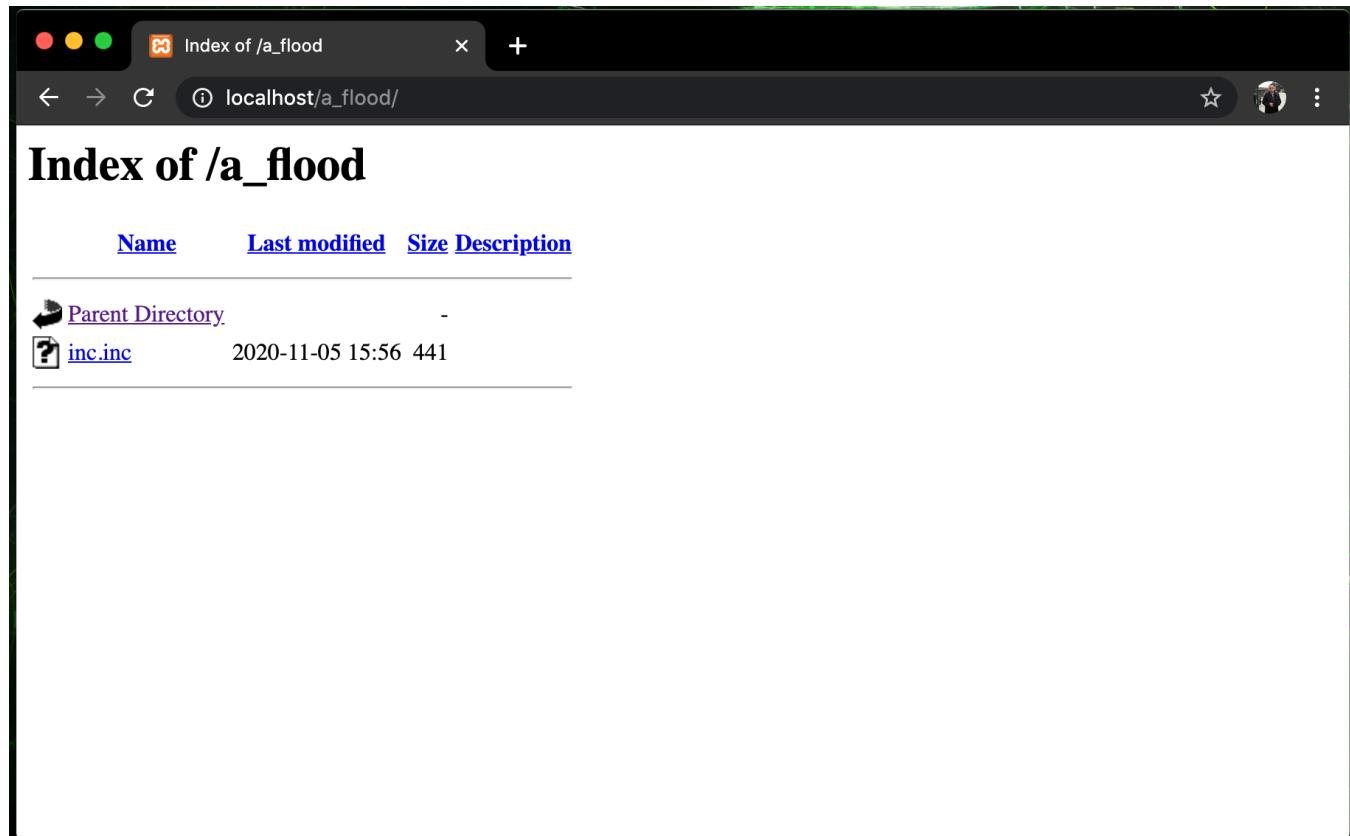


```
inc.inc      x      inc.inc      UNREGISTERED
1  <?php
2    $servername = "localhost";
3    $username = "root";
4    $password = "";
5    $dbname = "river_level_data"; //databasename
6    $tblname= "level_status_daerah1"; //table name
7    // Create connection
8    $conn = mysqli_connect($servername, $username,$password, $dbname );
9    // $connection = mysqli_select_db($conn,$dbname);

10   // Check connection
11   if (!$conn) {
12     die("Connection failed. " . mysqli_connect_error());
13   } else{
14     //echo "Connected to DB";
15   }
16 ?>
```

**Step 8:**

Test the **inc.inc** file by using any browser.



## Real time Flood Monitoring Database Example

### PHP, MySQLi, NodeMCU/ESP32 & Ultrasonic Sensor (HC-SR04)

---

Step 9:

Filename: **add.php** ← save inside c://xampp/htdocs/a\_flood/

**GET vs POST...**

```
<?php
include("inc.inc");

//Link=Connection();
date_default_timezone_set("Asia/Kuala_Lumpur");
$height=$_GET["dist"];// get data from NodeMCU & must same with in the NodeMCU variable name
$locate=$_GET["loc"];// get data from NodeMCU & must same with in the NodeMCU variable name

//Location
if ($locate == 1)
{
    $location = "SG PUSU, GOMBAK";
} else if ($locate ==2)
{
    $location = "SG CHINCHIN, GOMBAK";
}else{
    $location = "RIVER NOT LISTED!";
}

echo "<br>Height: ".$height. "cm & Location: ". $location; //flag to check data send form sensor

//THE FORMULA @ CONDITIONS
//only 2 conditions,
if ($height < 5) //if the height between sensor & water surface is less then 5cm-> status = DANGER
{
    $status = "DANGER";
} else {
    $status = "SAFE";
}

echo "<br>".$status;
//insert into table level_status_daerah1 (field from mysql) VALUES (data from php & NodeMCU)
$query = "INSERT INTO $tblname(river_name, water_level, river_status) VALUES ('$location','$height', '$status')";

if ((is_null($height) or is_null($location)))
{
    echo "Empty data"; //flag
}
else
{
    // if everything is complete, add data from NodeMCU inside the database
    mysqli_query($conn, $query) or die('Failed to update site table. Mysql returned the following:'.mysqli_error());
    echo "<br>Added to DB"; //flag
```

## Real time Flood Monitoring Database Example

### PHP, MySQLi, NodeMCU/ESP32 & Ultrasonic Sensor (HC-SR04)

```
}
```

```
//header("Location: index.php");
```

```
?>
```

Filename: add.php

The screenshot shows a code editor window titled "add.php". The code is a PHP script for a real-time flood monitoring system. It starts with a closing brace {}, followed by a header directive //header("Location: index.php");, and a closing tag ?>. The main body of the script handles data from a NodeMCU via GET requests for height and location. It sets the default timezone to Asia/Kuala\_Lumpur, checks the location (SG PUSU, GOMBAK or SG CHINCHIN, GOMBAK), and outputs a message indicating height and location. It then performs a conditional check based on height (< 5 cm) to determine a status (DANGER or SAFE). The script then inserts data into a MySQL database table named "level\_status\_daerah1" with columns river\_name, water\_level, and river\_status. If the insertion fails, it prints an error message. Finally, it outputs a message indicating the data was added to the database and ends with a header redirection. The code editor interface includes tabs for "add.php" and "UNREGISTERED", and status bars at the bottom showing "Line 46, Column 3", "Tab Size: 4", and "PHP".

```
add.php
```

```
4 // $link=Connection();
5 date_default_timezone_set("Asia/Kuala_Lumpur");
6 $height=$_GET["dist"]; // get data from NodeMCU & must same with in the NodeMCU port
7 $locate=$_GET["loc"]; // get data from NodeMCU & must same with in the NodeMCU port
8
9 //Location
10 if ($locate == 1)
11 {
12     $location = "SG PUSU, GOMBAK";
13 } else if ($locate ==2)
14 {
15     $location = "SG CHINCHIN, GOMBAK";
16 }else{
17     $location = "RIVER NOT LISTED!";
18 }
19 echo "<br>Height: ".$height. "cm & Location: ". $location; //flag to check data send form sensor
20
21 //THE FORMULA @ CONDITIONS
22 //only 2 conditions,
23 if ($height < 5) //if the height between sensor & water surface is less then 5cm-> status = DANGER
{
24     $status = "DANGER";
25 } else {
26     $status = "SAFE";
27 }
28
29 echo "<br>".$status;
30 //insert into table level_status_daerah1 (field from mysql) VALUES (data from php & NodeMCU)
31 $query = "INSERT INTO $tblname(river_name, water_level, river_status) VALUES ('$location','$height', '$status')";
32
33 if ((is_null($height) or is_null($location)))
{
34     echo "Empty data"; //flag
35 }
36 else
{
37     // if everything is complete, add data from NodeMCU inside the database
38     mysqli_query($conn, $query) or die('Failed to update site table. Mysql returned the following:' .mysqli_error());
39     echo "<br>Added to DB"; //flag
40 }
41 //header("Location: index.php");
42
43 ?>
```

Line 46, Column 3

Tab Size: 4

PHP

**Real time Flood Monitoring Database Example**  
**PHP, MySQLi, NodeMCU/ESP32 & Ultrasonic Sensor (HC-SR04)**

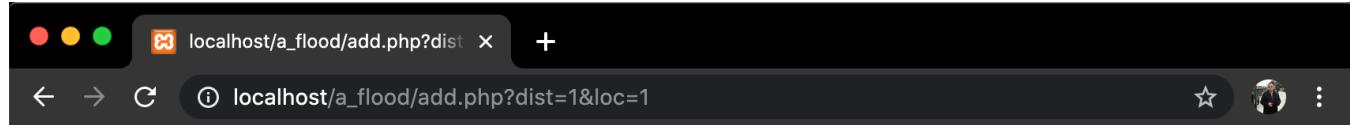
---

Step 10:

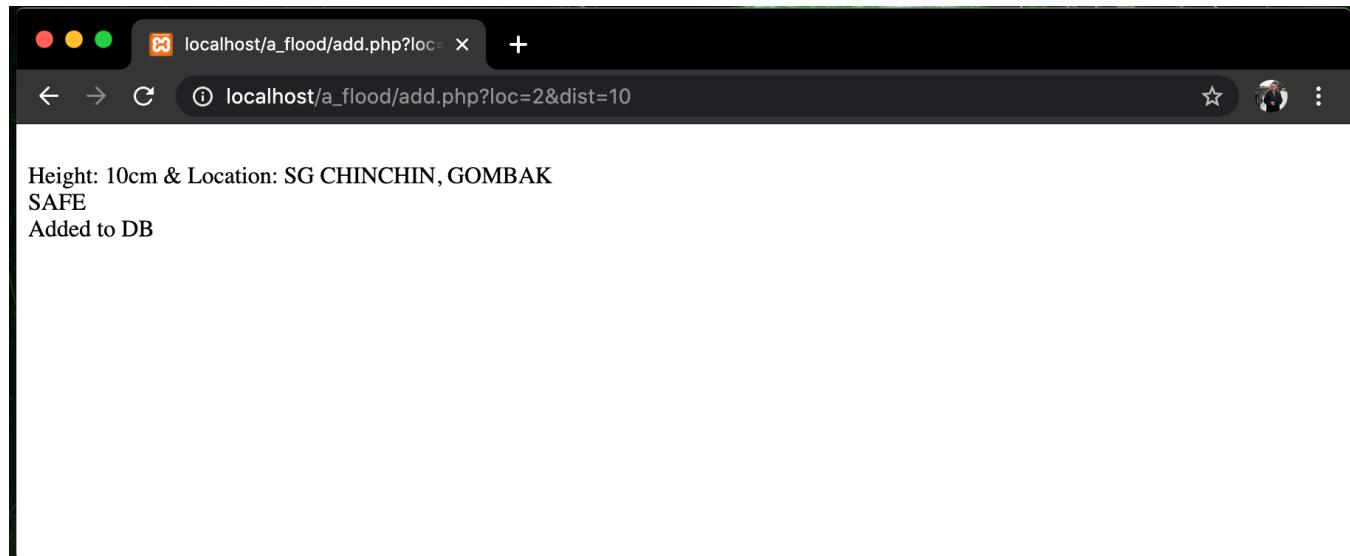
Test the database by injection to MySQL DB.

Have to troubleshoot the inc.inc & add.php files if unable to save into the database.

[http://localhost/a\\_flood/add.php?dist=1&loc=1](http://localhost/a_flood/add.php?dist=1&loc=1)



[http://localhost/a\\_flood/add.php?dist=10&loc=2](http://localhost/a_flood/add.php?dist=10&loc=2)



## Real time Flood Monitoring Database Example

### PHP, MySQLi, NodeMCU/ESP32 & Ultrasonic Sensor (HC-SR04)

#### Step 11:

Refresh the MySQL to view new inserted data.

The screenshot shows the phpMyAdmin interface for a MySQL database named 'river\_level\_data'. The current table is 'level\_status\_daerah1'. A yellow box highlights the last two rows of data, which were injected from the browser. An arrow points from the text 'The injected value from browser.' to the second row of the highlighted area. The table has columns: dataID, log, river\_name, water\_level, and river\_status. The injected rows contain dataID values 3 and 4, both with log entries from November 5, 2020, at 16:29:32. The river names are 'SG CHINCHIN, GOMBAK' and 'SG PUSU, GOMBAK' respectively. The water levels are 10 and 10, and the river statuses are 'SAFE' and 'SAFE'.

dataID	log	river_name	water_level	river_status
1	2020-11-05 14:06:51	river #1	level #1	status #1
2	2020-11-05 14:06:51	river #2	level #2	status #2
3	2020-11-05 16:29:32	SG CHINCHIN, GOMBAK	10	SAFE
4	2020-11-05 16:31:37	SG CHINCHIN, GOMBAK	10	SAFE
5	2020-11-05 16:53:33	SG PUSU, GOMBAK	2	DANGER
6	2020-11-05 16:54:26	SG PUSU, GOMBAK	1	DANGER

The injected value from browser.

## Real time Flood Monitoring Database Example

### PHP, MySQLi, NodeMCU/ESP32 & Ultrasonic Sensor (HC-SR04)

---

Step 12:

Filename: **index.php** ← save inside c://xampp/htdocs/a\_flood/

```
<?php

include("inc.inc");
$result=mysqli_query($conn,"SELECT * FROM $tblname ORDER BY dataID DESC"); //DESC menurun ASC menaik
?>
<html>
    <head>
        <title>Statistik Data Harian Sungai</title>
        <link rel="stylesheet" href=".css/style.css">
        <link rel="stylesheet" href=".css/bootstrap.css">
        <meta http-equiv="refresh" content="30"><!-- refresh every 30 seconds-->
    </head>
    <body>
        <div class="row">
            <div class="col-md-8 col-md-offset-2">
                <center><h3 style="text-align:right;" class="hijau tebel">STATUS KETINGGIAN AIR SUNGAI DI
DAERAH GOMBAK</h3></center>
            </div>
            <div class="col-md-2">
                &nbsp;
            </div>
        </div>
        <div class="row">
            <div class="col-md-8 col-md-offset-2">
                <center><h5 style="text-align:right;" class="miring">Data Logging dengan Arduino, NodeMCU &
Ultrasonic Sensor</h5></center>
                <hr style="margin-top: 0px; margin-bottom:0px">
            </div>
            <div class="col-md-2">
                &nbsp;
            </div>
        </div>
        <br>
        <div class="row">
            <div class="col-md-2 col-md-offset-2">
                <div class="panel panel-primary">
                    <div class="panel-heading">
                        <h3 class="panel-title tengah">Navigasi</h3>
                    </div>
                    <div class="panel-body" style="padding:0px;">
                        <table class="table table-striped table-hover" >
                            <tbody>
```

# Real time Flood Monitoring Database Example

## PHP, MySQLi, NodeMCU/ESP32 & Ultrasonic Sensor (HC-SR04)

## Real time Flood Monitoring Database Example

### PHP, MySQLi, NodeMCU/ESP32 & Ultrasonic Sensor (HC-SR04)

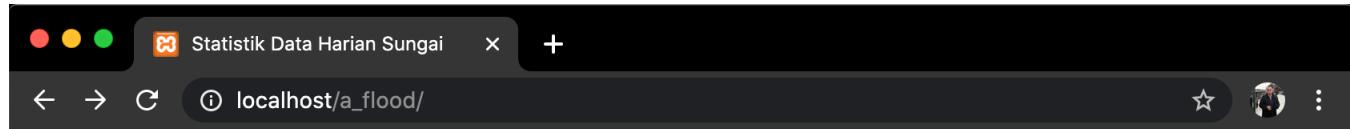
---

```
</div>
</div>
</body>
</html>

<script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js"></script>
<script type="text/javascript" src=".js/modules/data.js"></script>
<script type="text/javascript" src=".js/modules/exporting.js"></script>
<script type="text/javascript" src=".js/highcharts.js"></script>
<script type="text/javascript" src=".js/bootstrap.js"></script>
```

Output: [http://localhost/a\\_flood/](http://localhost/a_flood/) ← depends where you save the files.

**Real time Flood Monitoring Database Example**  
**PHP, MySQLi, NodeMCU/ESP32 & Ultrasonic Sensor (HC-SR04)**



## STATUS KETINGGIAN AIR SUNGAI DI DAERAH GOMBAK

*Data Logging dengan Arduino, NodeMCU & Ultrasonic Sensor*

**Navigasi**

- [Home](#)
- [Jadual/Table](#)
- [Statistik](#)

Tabel Data Suhu:

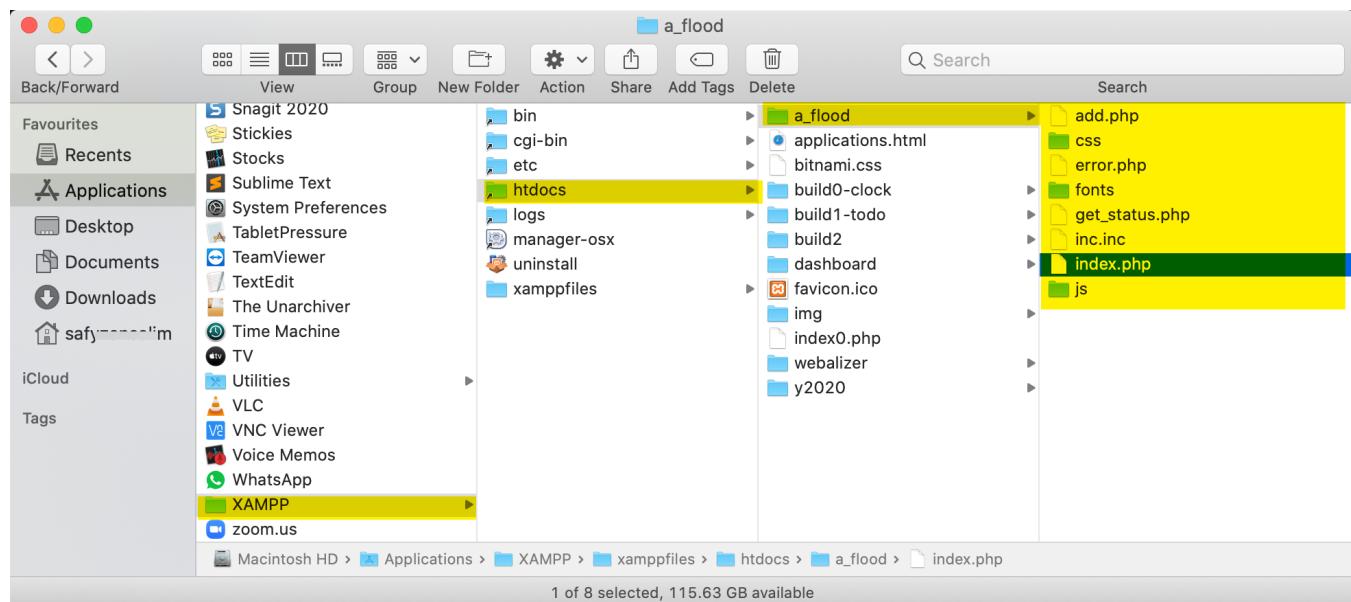
No	Tarikh Log	Nama Sungai	Level Sungai	Status Keadaan
8	2020-11-05 17:09:00	SG CHINCHIN, GOMBAK	10	SAFE
7	2020-11-05 17:08:55	SG PUSU, GOMBAK	1	DANGER
6	2020-11-05 16:54:26	SG PUSU, GOMBAK	1	DANGER
5	2020-11-05 16:53:33	SG PUSU, GOMBAK	2	DANGER
4	2020-11-05 16:31:37	SG CHINCHIN, GOMBAK	10	SAFE
3	2020-11-05 16:29:32	SG CHINCHIN, GOMBAK	10	SAFE
2	2020-11-05 14:06:51	river #2	level #2	status #2
1	2020-11-05 14:06:51	river #1	level #1	status #1

Location of files for this project.

C>>XAMPP>>htdocs>>a\_flood>>

## Real time Flood Monitoring Database Example

### PHP, MySQLi, NodeMCU/ESP32 & Ultrasonic Sensor (HC-SR04)



#### REMINDER.

- i. Make sure that your server / laptop and NodeMCU is under same network. It can be same hotspot, same access point. Otherwise NodeMCU cannot send the data to the server.
- ii. [http://localhost/a\\_flood/](http://localhost/a_flood/) only valid for local network – within the laptop itself. You need to get the laptop/server's IP address and type it at the NodeMCU's sketch.
- iii. You can find your ip add thru **cmd prompt > ipconfig**.
- iv. Do the database part, once ok, then you may concentrate on NodeMCU.
- v. If you unable to save the database change **`$_GET`** to **`$_POST`** in **`add.php`**  

```
$height=$_GET["dist"];// get data from NodeMCU & must same with in the NodeMCU port  
$locate=$_GET["loc"];// get data from NodeMCU & must same with in the NodeMCU port
```
- vi. Thank you & enjoy IoT.

**END OF PART B.**

#### C. Programming at NodeMCU side.

**Real time Flood Monitoring Database Example**  
**PHP, MySQLi, NodeMCU/ESP32 & Ultrasonic Sensor (HC-SR04)**

---

```
//UniKL BMI
//Connectthing Things & Big Data Analytics
//Jul 2019

//include libraries
#include <ESP8266HTTPClient.h>
#include <ESP8266WiFi.h>

// defines pins numbers
const int trigPin = D4;
const int echoPin = D3;

// defines variables
long duration;
int distance;

int delay_time = 3000; // Its mean 3 seconds wait to enter first record

//Access point credentials
const char* ssid      = "YES";://"IOT";
const char* pwd       = "a2856245";://"commtech123";

///String add_data_url = "http://192.168.43.20/iot-bda-szs/flood/add.php";
String add_data_url = "http://your-laptop-ip-add//a_flood/add.php";
WiFiServer server(80); // open port 80 for server connection

void setup() {
    pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
    pinMode(echoPin, INPUT); // Sets the echoPin as an Input

    Serial.begin(115200); //initialise the serial communication
    delay(20);
    WiFi.begin(ssid, pwd);

    //starting the server
    server.begin();
}
```

**Real time Flood Monitoring Database Example**  
**PHP, MySQLi, NodeMCU/ESP32 & Ultrasonic Sensor (HC-SR04)**

---

```
}

void loop() {
    delay(2000);
    WiFiClient client = server.available();
    // Clears the trigPin
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);

    // Sets the trigPin on HIGH state for 10 micro seconds
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    // Reads the echoPin, returns the sound wave travel time in microseconds
    duration = pulseIn(echoPin, HIGH);

    // Calculating the distance
    distance= duration*0.034/2;

    // Prints the distance on the Serial Monitor
    Serial.print("Distance: ");
    Serial.println(distance);

    /*if (distance <15)
    {
        Serial.println("ALERT!!!:");
    }else {
        Serial.println("SAFE!!!");
    }*/
    int loc = 1; //1 = Sg Pusu;
    add_device_data(distance,loc);
    delay(delay_time);
}

///replace this with your gps data
///ini adalah function
```

```
void add_device_data(int get_height, int get_location )  
{  
  
    WiFiClient client = server.available();  
  
    HTTPClient http;  
    ///// ?dist=... ni refer pada test database notes...  
    String url = add_data_url+"?dist="+get_height+"&loc="+get_location;  
  
    http.begin(url);  
    //GET method  
    int httpCode = http.GET();  
    String payload = http.getString();  
    Serial.println(url);  
    Serial.println(payload);  
    http.end();  
  
}  

```

**END OF PART C.**