/\* WEB Server for Head Circle counter

\*

\* Algoritma Lingkaran Ellipse bekerja dengan menghitung dua sumbu utama

\* dari lingkaran ellipse,yaitu sumbu utama horizontal (a) dan sumbu utama

\* Vertikal (b).Sumbu utama ini kemudian digunakan

\* untuk menghitung nilai lingkar kepala bayi.

\*

\* sensor depan 43,5 2580 5 417

\* sensor belakang 43,5 2479 5 346

\* sensor kanan 43,8 2500 5 296

\* sesnsor kiri 43,8 2509 5 431

\*/

#include <vector>

#include <WiFi.h>

#include <Preferences.h>

#include <LiquidCrystal\_I2C.h>

//Wi-Fi

const char\* ssid = "wifi-APLKB";

const char\* password = "12345678";

WiFiServer server(80);

String header;

// Preferences /EEPROM

Preferences preferences;

unsigned long ms\_current = 0;

unsigned long ms\_previous = 0;

const int depanTrig = 4;

const int depanEcho = 5;

const int belakangTrig = 13;

const int belakangEcho = 15;

const int kananTrig = 26;

const int kananEcho = 25;

const int kiriTrig = 19;

const int kiriEcho = 18;

const int tombolReset = 35;

// konfigurasi buzzer

const int speakerPin = 23;

const int notesLength = 2;

char notes[] = "C ";

int beats[] = {2, 1};

char notes1[] = "CCCC ";

int beats1[] = {1, 1, 1, 1, 1};

const int notes1Length = 5;

int tempo = 70;

// define sound speed in cm/us

#define SOUND\_SPEED 0.034

#define CM\_TO\_INCH 0.393701

const float konstanta = 44.5; // 43.5 & 43.7;

const float midKonstanta = konstanta \* 2/3;

const float phi = 3.14159;

// ukuran kotak = 43.5 cm

const float kaliF = 0; // 1.8

const float kaliB = 0; // 1.3

const float kaliR = 0; // 2

const float kaliL = 0; // 1.8

float jarakDepan;

float jarakBelakang;

float jarakKanan;

float jarakKiri;

float diaFB;

float diaRL;

float rataDia;

float lingkaran;

bool loopState = true;

LiquidCrystal\_I2C lcd(0x27, 16, 2);

// buat karakter

byte ArrowUp[8] = {

0b00000,

0b00100,

0b01110,

0b11111,

0b00100,

0b00100,

0b00100,

0b00000

};

byte ArrowDown[8] = {

0b00000,

0b00100,

0b00100,

0b00100,

0b11111,

0b01110,

0b00100,

0b00000

};

byte ArrowRight[8] = {

0b00000,

0b00100,

0b00110,

0b11111,

0b00110,

0b00100,

0b00000,

0b00000

};

byte ArrowLeft[8] = {

0b00000,

0b00100,

0b01100,

0b11111,

0b01100,

0b00100,

0b00000,

0b00000

};

void setup() {

Serial.begin(115200);

delay(50);

pinMode(depanTrig, OUTPUT); // Sets the depanTrig as an OUTPUT

pinMode(depanEcho, INPUT); // Sets the depanEcho as an INPUT

pinMode(belakangTrig, OUTPUT); // Sets the belakang Trig as an OUTPUT

pinMode(belakangEcho, INPUT); // Sets the belakang Echo as an INPUT

pinMode(kananTrig, OUTPUT); // Sets the kananTrig as an OUTPUT

pinMode(kananEcho, INPUT); // Sets the kananEcho as an INPUT

pinMode(kiriTrig, OUTPUT); // Sets the kiriTrig as an OUTPUT

pinMode(kiriEcho, INPUT); // Sets the kiriEcho as an INPUT

pinMode(speakerPin, OUTPUT);

pinMode(tombolReset, INPUT);

lcd.begin();

lcd.backlight();

lcd.createChar(0, ArrowUp);

lcd.createChar(1, ArrowDown);

lcd.createChar(2, ArrowRight);

lcd.createChar(3, ArrowLeft);

lcd.setCursor(0,0);

lcd.print("# Alat Ukur #");

lcd.setCursor(0,1);

lcd.print("#Lingkar Kepala#");

delay(2000);

lcd.clear();

startup\_wifi();

}

// function untuk buzzer

void playTone(int tone, int duration) {

for (long i = 0; i < duration \* 900L; i += tone \* 2)

{

digitalWrite(speakerPin, HIGH);

delayMicroseconds(tone);

digitalWrite(speakerPin, LOW);

delayMicroseconds(tone);

}

}

// function untuk buzzer

void playNote(char note, int duration) {

char names[] = { 'c', 'd', 'e', 'f', 'g', 'a', 'b', 'C' };

int tones[] = { 1915, 1700, 1519, 1432, 1275, 1136, 1014, 956 };

for (int i = 0; i < 8; i++) {

if (names[i] == note) {

playTone(tones[i], duration);

}

}

}

// function untuk bunyikan buzzer

void playBuzzer()

{

for (int i = 0; i < notesLength; i++)

{

if (notes[i] == ' ') {

delay(beats[i] \* tempo); // rest

} else {

playNote(notes[i], beats[i] \* tempo);

}

delay(tempo / 2);

}

}

void buzzerScan()

{

tempo = 70;

for (int i = 0; i < notesLength; i++) {

if (notes[i] == ' ') {

delay(beats[i] \* tempo); // rest

} else {

playNote(notes[i], beats[i] \* tempo);

}

delay(tempo / 2);

}

}

void buzzerObject()

{

tempo = 70;

for (int i = 0; i < notes1Length; i++) {

if (notes1[i] == ' ') {

delay(beats1[i] \* tempo); // rest

} else {

playNote(notes1[i], beats1[i] \* tempo);

}

delay(tempo / 2);

}

}

void buzzerScanDone()

{

tempo = 300;

for (int i = 0; i < notesLength; i++) {

if (notes[i] == ' ') {

delay(beats[i] \* tempo); // rest

} else {

playNote(notes[i], beats[i] \* tempo);

}

delay(tempo / 2);

}

}

float tes\_ukur()

{

jarakDepan = tes\_sensor(depanTrig, depanEcho, 0) - 2;

delay(100);

jarakBelakang = tes\_sensor(belakangTrig, belakangEcho, 1) - 0.9;

delay(100);

jarakKanan = tes\_sensor(kananTrig, kananEcho, 2) - 1;

delay(100);

jarakKiri = tes\_sensor(kiriTrig, kiriEcho, 3) - 1;

delay(100);

// menampilkan pada terminal

Serial.print(" ");

Serial.println(jarakDepan, 1);

Serial.print(jarakKiri, 1);

Serial.print(" ");

Serial.println(jarakKanan, 1);

Serial.print(" ");

Serial.println(jarakBelakang, 1);

// jika ada objek, maka data sensor diolah

if (jarakDepan < midKonstanta && jarakBelakang < midKonstanta && jarakKanan < midKonstanta && jarakKiri < midKonstanta)

{

buzzerObject();

lcd.clear();

lcd.setCursor(0,0);

lcd.print(" ...geser...");

lcd.setCursor(0,1);

if ( jarakDepan < jarakBelakang - 1 )

{

lcd.print(" ");

lcd.write(0);

lcd.write(0);

lcd.write(0);

lcd.write(0);

lcd.write(0);

lcd.print(" ");

}

else if ( jarakDepan - 1 > jarakBelakang )

{

lcd.print(" ");

lcd.write(1);

lcd.write(1);

lcd.write(1);

lcd.write(1);

lcd.write(1);

lcd.print(" ");

}

else if ( jarakKanan - 1 > jarakKiri )

{

lcd.print(" ");

lcd.write(3);

lcd.write(3);

lcd.write(3);

lcd.write(3);

lcd.write(3);

lcd.print(" ");

}

else if ( jarakKanan < jarakKiri - 1 )

{

lcd.print(" ");

lcd.write(2);

lcd.write(2);

lcd.write(2);

lcd.write(2);

lcd.write(2);

lcd.print(" ");

}

// jika

// sensor depan & sensor belakang selisih di bawah 2 cm,

// sensor kanan & sensor kiri selisih di bawah 2 cm,

// maka ready ambil data

float selFB = jarakDepan - jarakBelakang;

float selRL = jarakKanan - jarakKiri;

if ( selFB < 0 ) selFB = selFB \* -1;

if ( selRL < 0 ) selRL = selRL \* -1;

if ( selFB < 2 && selRL < 2 )

{

// olah data dari sensor

diaFB = konstanta - ( jarakDepan + jarakBelakang );

diaRL = konstanta - ( jarakKanan + jarakKiri );

rataDia = ( diaFB + diaRL ) / 2;

// rumus lingkaran

lingkaran = phi \* rataDia;

lcd.clear();

// Prints the distance in the Serial Monitor

Serial.print("diameter FB: ");

Serial.println(diaFB);

Serial.print("diameter RL: ");

Serial.println(diaRL);

Serial.print("diameter lingkaran: ");

Serial.println(rataDia);

Serial.println("===");

Serial.println(lingkaran, 1);

Serial.println("===");

// tampilkan data ke LCD

lcd.setCursor(0,0);

lcd.print("D1:");

lcd.print(diaFB, 1);

lcd.setCursor(9,0);

lcd.print("D2:");

lcd.print(diaRL, 1);

lcd.setCursor(0,1);

lcd.print("L.Kepala:");

lcd.print(lingkaran, 2);

lcd.print("cm");

loopState = true;

for (byte x = 0; x < 15; x++ ){

int counterDelay = 0;

lcd.noBacklight();

do {

if ( digitalRead(tombolReset) == 1 ) loopState = false;

if ( loopState == false ) break;

delay(1);

counterDelay++;

// Serial.print(digitalRead(tombolReset));

} while( counterDelay <= 200 );

lcd.backlight();

buzzerScanDone();

if ( loopState == false ) break;

do {

if ( digitalRead(tombolReset) == 1 ) loopState = false;

if ( loopState == false ) break;

delay(1);

counterDelay++;

// Serial.print(digitalRead(tombolReset));

} while( counterDelay <= 800 );

cek\_client();

}

lcd.clear();

lcd.setCursor(0,0);

lcd.print(" Scanning...");

delay(2000);

//reset nilai hasil pengukuran

diaFB = 0;

diaRL = 0;

rataDia = 0;

lingkaran = 0;

}

}

else {

buzzerScan();

Serial.println("objek tidak terdeteksi...");

lcd.clear();

lcd.setCursor(0,0);

lcd.print(" Scanning...");

lcd.setCursor(0,1);

lcd.print(" No Object ");

}

}

float tes\_sensor(int trig, int echo, int pos)

{

float duration;

float jarakCm;

float jarakInch;

String posisi[] = {"depan","belakang","kanan","kiri"};

// Clears the Trig

digitalWrite(trig, LOW);

delayMicroseconds(2);

// Sets the Trig in HIGH state for 10 micro seconds

digitalWrite(trig, HIGH);

delayMicroseconds(10);

digitalWrite(trig, LOW);

// Reads the Echo, returns the sound wave travel time in microseconds

duration = pulseIn(echo, HIGH);

delay(20);

// Serial.print("duration ");

// Serial.print(posisi[pos]);

// Serial.print(" : ");

// Serial.println(duration);

// Calculate the distance in cm

// jarakCm = duration \* SOUND\_SPEED/2;

jarakCm = duration / 58;

// Convert to inches if needed

jarakInch = jarakCm \* CM\_TO\_INCH;

switch(pos)

{

case 0:

// Serial.println("ini sensor depan");

// if (duration > 200 && duration < 450){

// jarakCm = jarakCm + 0.25;

// } else if (duration >= 680 && duration < 850 ) {

// jarakCm = jarakCm - 0.3;

// } else if (duration >= 1000 && duration < 1160){

// jarakCm = jarakCm - 0.7;

// } else if (duration >= 1160){

// jarakCm = jarakCm - 0.4;

// }

break;

case 1:

// Serial.println("ini sensor belakang");

// if (duration > 200 && duration < 360){

// jarakCm = jarakCm + 0.4;

// } else if (duration >= 360 && duration < 475){

// jarakCm = jarakCm + 0.5;

// } else if (duration >= 475 && duration > 600){

// jarakCm = jarakCm + 0.6;

// } else if (duration >= 600 && duration < 650){

// jarakCm = jarakCm + 0.2;

// } else if (duration >= 650 && duration < 700){

// jarakCm = jarakCm + 0.6;

// } else if (duration >= 700 && duration < 1040){

// jarakCm = jarakCm + 0.3;

// } else if (duration >= 1040){

// jarakCm = jarakCm + 0.4;

// }

break;

case 2:

// Serial.println("ini sensor kanan");

// if (duration > 200 && duration < 490){

// jarakCm = jarakCm + 0.3;

// } else if (duration >= 530 && duration < 610){

// jarakCm = jarakCm - 0.4;

// }

// else if (duration >= 700 && duration < 740){

// jarakCm = jarakCm + 0.5;

// } else if (duration >= 740 && duration < 950){

// jarakCm = jarakCm - 0.25;

// }

// else if (duration >= 950 && duration < 1215){

// jarakCm = jarakCm + 0.25;

// } else if (duration >= 1215){

// jarakCm = jarakCm + 0.5;

// }

break;

case 3:

// Serial.println("ini sensor kiri");

// if (duration > 200 && duration < 430){

// jarakCm = jarakCm + 0.3;

// } else if (duration >= 430 && duration < 510){

// jarakCm = jarakCm - 0.4;

// } else if (duration >= 540 && duration < 1100){

// jarakCm = jarakCm + 0.6;

// } else if (duration >= 1100 && duration < 1220){

// jarakCm = jarakCm - 0.1;

// } else if (duration >= 1220){

// jarakCm = jarakCm + 0.3;

// }

break;

default:

break;

}

return jarakCm;

}

float sensor\_baca(int trig, int echo, int mVal, int cVal)

{

float duration;

float jarakCm;

float jarakInch;

std::vector<long> durSensor = {};

for (int i = 0; i < 1; i++)

{

// Clears the Trig

digitalWrite(trig, LOW);

delayMicroseconds(2);

// Sets the Trig in HIGH state for 10 micro seconds

digitalWrite(trig, HIGH);

delayMicroseconds(10);

digitalWrite(trig, LOW);

// Reads the Echo, returns the sound wave travel time in microseconds

duration = pulseIn(echo, HIGH);

// duration = duration + kalibrasi;

durSensor.push\_back(duration);

delay(20);

}

int durSensorCount = sizeof(durSensor) / sizeof(durSensor[0]);

int maxCount = 0;

int modeValue;

int nilaiSensor = 0;

for (int j = 0; j < durSensor.size(); j++ )

{

nilaiSensor = nilaiSensor + durSensor[j];

}

Serial.print("duration: ");

Serial.println(duration);

// Calculate the distance in cm

jarakCm = duration \* SOUND\_SPEED/2;

// jarakCm = duration / 58;

// kalau 417 = 5 cm , kalau 2580 = 43,5 cm , kalau 1000 berarti

// A = ( B - c ) / m

// jarakCm = ( duration - cVal ) / mVal;

// Convert to inches if needed

jarakInch = jarakCm \* CM\_TO\_INCH;

return jarakCm;

}

void ukur()

{

int mDepan, cDepan;

int mBelakang, cBelakang;

int mKanan, cKanan;

int mKiri, cKiri;

int delaySensor = 50; //stable: 100

// mDepan = (2524 - 331) / (konstanta - 5);

// cDepan = 331 - (mDepan \* 5);

// mBelakang = (2500 - 295) / (konstanta - 5);

// cBelakang = 295 - (mBelakang \* 5);

// mKanan = (2545 - 296) / (konstanta - 5);

// cKanan = 296 - (mKanan \* 5);

// mKiri = (2531 - 346) / (konstanta - 5);

// cKiri = 346 - (mKiri \* 5);

mDepan = (1937 - 348) / (33.5 - 5);

cDepan = 348 - (mDepan \* 5);

mBelakang = (1925 - 346) / (33.5 - 5);

cBelakang = 346 - (mBelakang \* 5);

mKanan = (1929 - 312) / (34 - 5);

cKanan = 312 - (mKanan \* 5);

mKiri = (1966 - 279) / (34 - 5);

cKiri = 279 - (mKiri \* 5);

jarakDepan = sensor\_baca(depanTrig, depanEcho, mDepan , cDepan );

jarakDepan = jarakDepan + kaliF;

delay(delaySensor);

jarakBelakang = sensor\_baca(belakangTrig, belakangEcho, mBelakang , cBelakang);

jarakBelakang = jarakBelakang + kaliB;

delay(delaySensor);

jarakKiri = sensor\_baca(kiriTrig, kiriEcho, mKiri , cKiri);

jarakKiri = jarakKiri + kaliL;

delay(delaySensor);

jarakKanan = sensor\_baca(kananTrig, kananEcho, mKanan , cKanan);

jarakKanan = jarakKanan + kaliR;

delay(delaySensor);

// menampilkan pada terminal

Serial.print(" ");

Serial.println(jarakDepan, 1);

Serial.print(jarakKiri, 1);

Serial.print(" ");

Serial.println(jarakKanan, 1);

Serial.print(" ");

Serial.println(jarakBelakang, 1);

// jika ada objek, maka data sensor diolah

if (jarakDepan < midKonstanta && jarakBelakang < midKonstanta && jarakKanan < midKonstanta && jarakKiri < midKonstanta)

{

// lcd.clear();

// lcd.setCursor(0,0);

// lcd.print(" ...geser...");

// lcd.setCursor(0,1);

// if ( jarakDepan < jarakBelakang - 1 )

// {

// lcd.print(" ");

// lcd.write(0);

// lcd.write(0);

// lcd.write(0);

// lcd.write(0);

// lcd.write(0);

// lcd.print(" ");

// }

// else if ( jarakDepan - 1 > jarakBelakang )

// {

// lcd.print(" ");

// lcd.write(1);

// lcd.write(1);

// lcd.write(1);

// lcd.write(1);

// lcd.write(1);

// lcd.print(" ");

// }

// else if ( jarakKanan - 1 > jarakKiri )

// {

// lcd.print(" ");

// lcd.write(3);

// lcd.write(3);

// lcd.write(3);

// lcd.write(3);

// lcd.write(3);

// lcd.print(" ");

// }

// else if ( jarakKanan < jarakKiri - 1 )

// {

// lcd.print(" ");

// lcd.write(2);

// lcd.write(2);

// lcd.write(2);

// lcd.write(2);

// lcd.write(2);

// lcd.print(" ");

// }

//

//

// // jika

// // sensor depan & sensor belakang selisih di bawah 2 cm,

// // sensor kanan & sensor kiri selisih di bawah 2 cm,

// // maka ready ambil data

//

// float selFB = jarakDepan - jarakBelakang;

// float selRL = jarakKanan - jarakKiri;

// if ( selFB < 0 ) selFB = selFB \* -1;

// if ( selRL < 0 ) selRL = selRL \* -1;

//

//

//// if ( selFB < 2 && selRL < 2 )

//// { // 483

// // olah data dari sensor

// diaFB = 43.21 - ( jarakDepan + jarakBelakang ); // jarak sensor depan ke belakang = 43,21

// diaRL = 43,7 - ( jarakKanan + jarakKiri ); // jarak sensor kanan ke kiri = 43,7

// rataDia = ( diaFB + diaRL ) / 2;

//// rataDia = sqrt( 2 \* pow(jarakDepan, 2) + 2 \* pow(jarakBelakang, 2) + 2 \* pow(jarakKanan, 2) + 2 \* pow(jarakKiri, 2 ) );

//

//

//// Rumus Lingkar Kepala = π × √(2 × a^2 + 2 × b^2)

//// lingkaran = phi \* sqrt( ( 2 \* pow(diaFB, 2) + 2 \* pow(diaRL, 2) ) );

//

//

// lingkaran = phi \* rataDia;

//

// lcd.clear();

//

// // Prints the distance in the Serial Monitor

// Serial.print("diameter FB: ");

// Serial.println(diaFB);

// Serial.print("diameter RL: ");

// Serial.println(diaRL);

// Serial.print("diameter lingkaran: ");

// Serial.println(rataDia);

//

//

// Serial.println("===");

// Serial.println(lingkaran, 1);

// Serial.println("===");

//

//// tampilkan data ke LCD

// lcd.setCursor(0,0);

// lcd.print("D1:");

// lcd.print(diaFB, 1);

//

// lcd.setCursor(9,0);

// lcd.print("D2:");

// lcd.print(diaRL, 1);

//

// lcd.setCursor(0,1);

// lcd.print("L.Kepala:");

// lcd.print(lingkaran, 1);

// lcd.print(" cm");

//

// for (byte x = 0; x < 30; x++ ){

// bool loopState = true;

// int counterDelay = 0;

//

// lcd.noBacklight();

// cek\_client();

//

//

// do {

// if ( digitalRead(tombolReset) == 1 ) loopState = false;

// if ( loopState == false ) break;

// delay(1);

// counterDelay++;

// } while( counterDelay <= 200 );

//

// lcd.backlight();

// if ( loopState == false ) break;

//

// cek\_client();

//

// do {

// if ( digitalRead(tombolReset) == 1 ) loopState = false;

// if ( loopState == false ) break;

// delay(1);

// counterDelay++;

// } while( counterDelay <= 1000 );

//

// }

//

// lcd.clear();

// lcd.setCursor(0,0);

// lcd.print(" Scanning...");

// delay(2000);

//

// //reset nilai hasil pengukuran

// diaFB = 0;

// diaRL = 0;

// rataDia = 0;

// lingkaran = 0;

// }

}

else {

Serial.println("objek tidak terdeteksi...");

lcd.clear();

lcd.setCursor(0,0);

lcd.print(" Scanning...");

lcd.setCursor(0,1);

lcd.print(" No Object ");

}

}

void startup\_wifi()

{

Serial.print("Setting AP (Access Point)...");

WiFi.softAP(ssid, password);

IPAddress IP = WiFi.softAPIP();

Serial.print("AP IP address: ");

Serial.println(IP);

server.begin();

lcd.setCursor(0,0);

lcd.print(" scanning...");

}

void cek\_client()

{

byte counterWIFI = 0;

WiFiClient client = server.available();

if (client) { // If a new client connects,

Serial.println("New Client."); // print a message out in the serial port

String currentLine = ""; // make a String to hold incoming data from the client

long counterWhile = 0;

while (client.connected()) { // loop while the client's connected

if (client.available()) { // if there's bytes to read from the client,

char c = client.read(); // read a byte, then

// Serial.write(c); // print it out the serial monitor

header += c;

if (c == '\n') { // if the byte is a newline character

// if the current line is blank, you got two newline characters in a row.

// that's the end of the client HTTP request, so send a response:

if (currentLine.length() == 0) {

// getting the GET method parameter from client using substring();

int ix1 = header.indexOf("GET ");

int ix2 = header.indexOf("HTTP/1.1");

String parameterGET = header.substring(ix1 + 3 , ix2);

Serial.print("index awal: ");

Serial.println(ix1);

Serial.print("index akhir: ");

Serial.println(ix2);

Serial.print("isi GET requestnya: ");

Serial.println(parameterGET);

if ( parameterGET.indexOf("scanUlang") > 0 )

{

Serial.println("perintah scan ulang...");

loopState = false;

diaFB = 0;

diaRL = 0;

rataDia = 0;

lingkaran = 0;

}

// HTTP headers always start with a response code (e.g. HTTP/1.1 200 OK)

// and a content-type so the client knows what's coming, then a blank line:

client.println("HTTP/1.1 200 OK");

client.println("Content-type:text/html");

client.println("Connection: close");

client.println();

// ukur();

// Display the HTML web page

client.println("<!DOCTYPE html><html>");

client.println("<head><meta name=\"viewport\" content=\"width=device-width, initial-scale=1\">");

client.println("<link rel=\"icon\" href=\"data:,\">");

client.println("<title>Pengukuran Lingkar Kepala Bayi</title>");

// CSS to style the on/off buttons

// Feel free to change the background-color and font-size attributes to fit your preferences

client.println("<style>\*{padding:0; margin:0;}");

client.println("html { background-image:linear-gradient(to right, #eee, #ddd); font-family: Helvetica; display: inline-block; margin: 0px auto; text-align: center;}");

client.println("h2 {margin: 20px 0 10px 0;}");

client.println("h4 {margin: 20px 0 10px 0;}");

client.println("h5 {text-align: center;background-color:#1fd31f;border-bottom:solid black 1px; padding:10px 0 7px 0; width:90%; margin:auto; border-radius: 10px 10px 0 0; }");

client.println(".button { border-radius: 8px; background-color: #2c2591; border: none; color: white; padding: 12px 35px;");

client.println("text-decoration: none; font-size: 14px; font-weight: bold; margin: 10px; cursor: pointer;}");

client.println(".button:hover {background-color:rgb(139,102,197); color:black;}");

client.println(".button1 {background-color: #FF1111;}");

client.println(".button2 {background-color: #555555;}");

client.println("input { border-radius: 8px; background-color: #eee; width: 20%; padding: 10px; margin:10px 0 10px 0; text-align: center;}");

client.println(".container{padding: 5px;display: flex;flex-direction: row; align-items: center; justify-content: center;flex-wrap: wrap;}");

client.println(".container hr{width: 50%; margin: auto;} footer{font-size: 10px; margin: 20px 0;}");

client.println("#lingkaran{margin-top:20px; margin-bottom:20px; font-weight: bold; font-size:large; background-color:#ff4300;}"); //ff4300 5dd6e5

client.println("h5.hasil{background-color:#4caf50;}");

client.println("</style></head>");

// Web Page Heading

client.println("<body><h2>PENGUKURAN LINGKAR KEPALA</h2>");

client.println("<hr>");

client.println("<h4>Data Sensor Ultra Sonik</h4>");

client.println("<form method=\"GET\" action=\"\">");

client.println(" <div class=\"container\">");

client.println(" <div class=\"content\">");

client.println(" <h5>Sensor Depan</h5>");

client.println(" <input type=\"text\" name=\"sensorF\" value=\"" + String(jarakDepan) + "\" >");

client.println(" </div>");

client.println(" <div class=\"content\">");

client.println(" <h5>Sensor Belakang</h5>");

client.println(" <input type=\"text\" name=\"sensorB\" value=\"" + String(jarakBelakang) + "\" >");

client.println(" </div>");

client.println(" <div class=\"content\">");

client.println(" <h5>Sensor Kanan</h5>");

client.println(" <input type=\"text\" name=\"sensorR\" value=\"" + String(jarakKanan) + "\" >");

client.println(" </div>");

client.println(" <div class=\"content\">");

client.println(" <h5>Sensor Kiri</h5>");

client.println(" <input type=\"text\" name=\"sensorL\" value=\"" + String(jarakKiri) + "\" >");

client.println(" </div>");

client.println(" </div>");

client.println(" <br>");

client.println(" <hr>");

client.println(" <div class=\"container\">");

client.println(" <div class=\"content\">");

client.println(" <h5>Diamater Depan Belakang</h5>");

client.println(" <input type=\"text\" name=\"diaFB\" value=\"" + String(diaFB) + "\" >");

client.println(" </div>");

client.println(" <div class=\"content\">");

client.println(" <h5>Diamater Kanan Kiri</h5>");

client.println(" <input type=\"text\" name=\"diaRL\" value=\"" + String(diaRL) + "\" >");

client.println(" </div>");

client.println(" <div class=\"content\">");

client.println(" <h5>Diameter Rata-Rata</h5>");

client.println(" <input type=\"text\" name=\"rataDia\" value=\"" + String(rataDia) + "\" >");

client.println(" </div>");

client.println(" </div>");

client.println(" <div class=\"container\">");

client.println(" <div class=\"content\">");

client.println(" <h5>Ukuran Lingkaran Kepala</h5>");

client.println(" <input type=\"text\" id=\"lingkaran\" name=\"lingkaran\" value=\"" + String(lingkaran) + "\" >");

client.println(" <br><button id=\"tombolData\" class=\"button\">Ambil Data</button>");

client.println(" </div>");

client.println(" </div>");

client.println("</form>");

client.println("<form method=\"GET\" action=\"\">");

client.println(" <button id=\"tombolScan\" name=\"tombolScan\" value=\"scanUlang\" class=\"button button1\">Scan Ulang</button>");

client.println("</form>");

client.println("<hr><footer><p class=\"copyright\"> <i>Alat Pengukur Lingkar Kepala Bayi</i> &copy; 2023</p></footer>");

client.println("<script>");

// client.println("let waktuReload = 2000;");

client.println("function pesan(){location.href = \"http://192.168.4.1\";}");

client.println("const lingkaran = document.querySelector(\"#lingkaran\");");

client.println("lingkaran.style.backgroundColor = \"#5dd6e5\"; setTimeout( pesan , 10000 );;");

client.println("if (lingkaran.value === \"0.00\") {lingkaran.style.backgroundColor = \"#ff4300\"; setTimeout( pesan , 2000 );} ");

client.println("</script>");

client.println("</body></html>");

// The HTTP response ends with another blank line

client.println();

// Break out of the while loop

break;

}

else

{ // if you got a newline, then clear currentLine

currentLine = "";

}

} else if (c != '\r') { // if you got anything else but a carriage return character,

currentLine += c; // add it to the end of the currentLine

}

}

// ketika terhubung ke aplikasi melalui HP, program terjebak di while loop

// counterWhile digunakan agar mengatasi hal tersebut

counterWhile++;

if (counterWhile % 1000 == 0) Serial.println(counterWhile);

if ( digitalRead(tombolReset) == 1 || counterWhile > 1000 ){

client.stop();

Serial.println("Client disconnected.");

}

}

// Serial.println(header);

// Clear the header variable

header = "";

// Close the connection

client.stop();

Serial.println("Client disconnected.");

Serial.println("");

}

}

void loop() {

ms\_current = millis();

cek\_client();

if ( ms\_current - ms\_previous >= 2000)

{

ms\_previous = ms\_current;

tes\_ukur();

}

}