TUGAS SIB ARKATAMA MULTI SOLUSINDO  
IOT PRAKTIK

ID Kegiatan : 7582873

Nama : Evy Nur Imamah

Kelas : IoT 1

**TOOLS!**

1. <https://wokwi.com/>

**Soal!**

1. Buatlah proyek dengan skema berikut :

A circuit board with wires

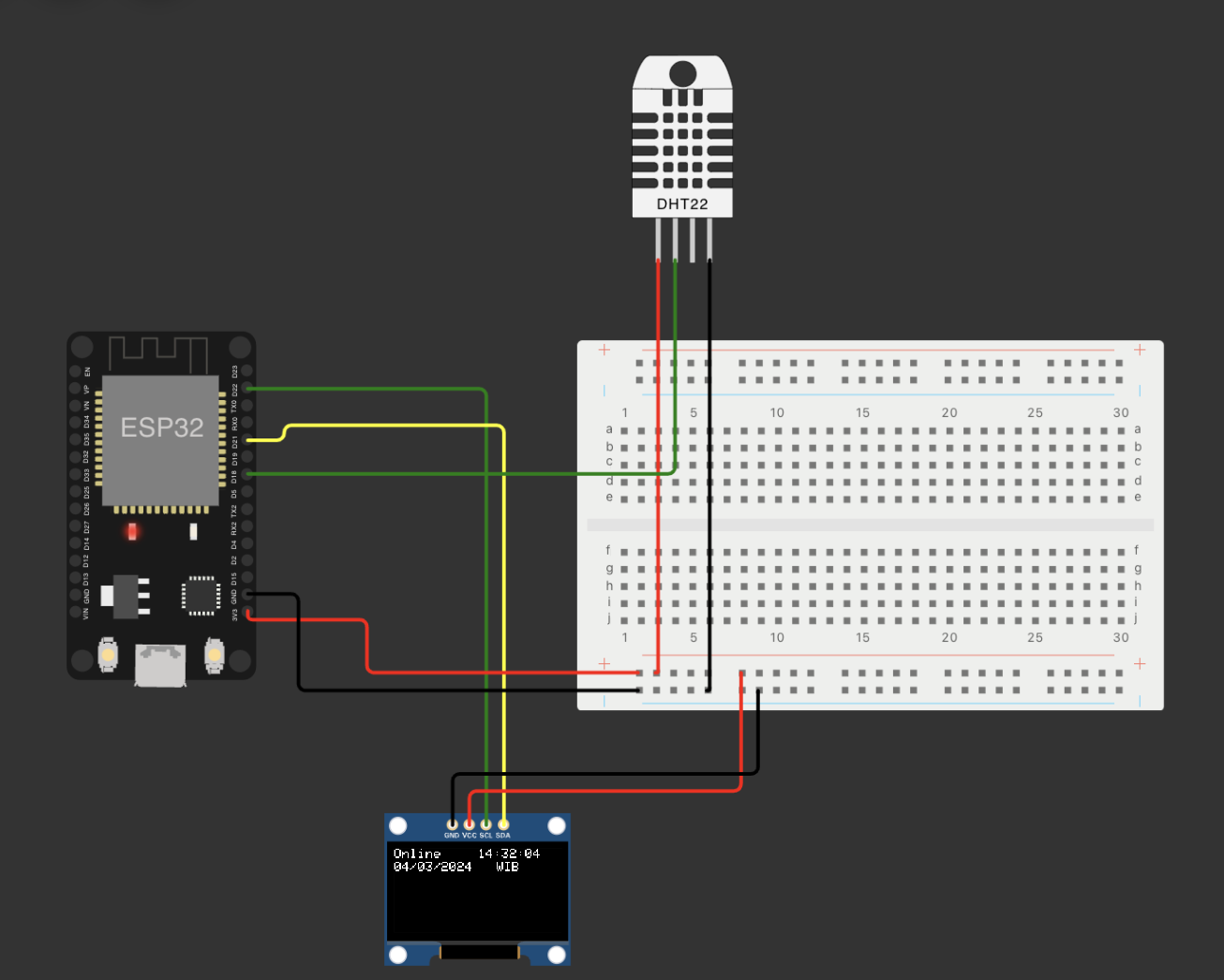
Description automatically generated

A circuit board with wires

Description automatically generated

A circuit board with wires

Description automatically generated



Deskripsi Proyek:

Proyek ini bertujuan untuk membuat sebuah perangkat IoT yang dapat menampilkan identitas pengguna (nama, NIM, dan kelas IoT) serta menampilkan waktu yang disinkronkan dengan server NTP pada layar OLED. Proyek ini menggunakan mikrokontroler ESP32 untuk mengelola koneksi WiFi dan proses sinkronisasi waktu dengan server NTP.

Spesifikasi Proyek:

* 1. Identitas Pengguna:
* Aplikasi ini akan menampilkan informasi identitas pengguna berupa nama, NIM, dan kelas IoT. Informasi ini akan direpresentasikan dalam bentuk teks yang jelas dan mudah dibaca.
  1. Koneksi WiFi:
* Mikrokontroler ESP32 akan dihubungkan dengan jaringan WiFi. Jaringan WiFi yang digunakan dalam proyek ini adalah "wokwi". Proses koneksi ke jaringan WiFi akan dilakukan secara otomatis setelah perangkat dinyalakan.
  1. Sinkronisasi NTP:
* Perangkat akan melakukan sinkronisasi waktu dengan server NTP (Network Time Protocol) untuk memastikan bahwa waktu yang ditampilkan adalah akurat. Sinkronisasi waktu akan dilakukan secara berkala untuk memperbaharui waktu pada perangkat.
  1. Tampilan OLED:
* Informasi identitas pengguna dan waktu yang telah disinkronkan akan ditampilkan pada layar OLED. Layar OLED akan menampilkan teks dengan kontras yang baik sehingga mudah dibaca, bahkan dalam kondisi pencahayaan rendah.

Langkah-langkah Proyek:

1. Pengaturan Perangkat:

* Siapkan semua perangkat yang diperlukan, termasuk mikrokontroler ESP32, layar OLED, dan koneksi WiFi "wokwi".

1. Program Mikrokontroler:

* Program harus mencakup pengaturan koneksi WiFi, sinkronisasi waktu dengan server NTP, dan logika untuk menampilkan informasi identitas pengguna dan waktu pada layar OLED.

1. Konversi Suhu:

* Implementasikan rumus konversi suhu dari Celcius ke Fahrenheit dan Kelvin dalam kode program.

1. Testing:

* Perhatikan output pada layar OLED dan pastikan bahwa semua informasi ditampilkan dengan benar.

**Jawab!**

1. Rangkaian Skematik (\*gambar/screenshoot)

|  |
| --- |
| A circuit board with wires  Description automatically generated |

1. Program (.ino,.c)

|  |
| --- |
| #include <Wire.h>  #include <Adafruit\_GFX.h>  #include <Adafruit\_SSD1306.h>  #include <DHT.h>  #include <WiFi.h>  #include <NTPClient.h>  #include <TimeLib.h>  #define SCREEN\_WIDTH 128 // OLED display width, in pixels  #define SCREEN\_HEIGHT 64 // OLED display height, in pixels  #define OLED\_RESET -1    // Reset pin # (or -1 if sharing Arduino reset pin)  #define DHTPIN 18        // Pin which is connected to the DHT sensor  #define DHTTYPE DHT22    // DHT 22  (AM2302)  #define NAMA "Evy Nur Imamah"  #define NIM "210411100123"  #define KELAS "IoT 1"  const char \*ssid = "Wokwi-GUEST"; // WiFi SSID  const char \*ntpServer = "pool.ntp.org";  const long gmtOffset\_sec = 25200; // Waktu Indonesia Barat (WIB) offset, 7 jam dari UTC  const int daylightOffset\_sec = 0;  DHT dht(DHTPIN, DHTTYPE);  Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &**Wire**, OLED\_RESET);  WiFiUDP ntpUDP;  NTPClient timeClient(ntpUDP, ntpServer, gmtOffset\_sec, daylightOffset\_sec);  void setup() {  **Serial**.begin(9600);  if (!display.begin(SSD1306\_SWITCHCAPVCC, 0x3C)) {  **Serial**.println(F("SSD1306 allocation failed"));  for (;;);  }  display.display();  delay(2000); // Pause for 2 seconds  display.clearDisplay();  display.setTextSize(1);  display.setTextColor(SSD1306\_WHITE);  display.setCursor(0,0);  display.print("Nama: ");  display.println(NAMA);  display.print("NIM: ");  display.println(NIM);  display.print("Kelas: ");  display.println(KELAS);  display.display();  dht.begin();  WiFi.begin(ssid, "", 6);  while (WiFi.status() != WL\_CONNECTED) {  delay(1000);  **Serial**.print(".");  }  **Serial**.println("");  **Serial**.println("WiFi connected");  **Serial**.print("IP address: ");  **Serial**.println(WiFi.localIP());  **Serial**.println("Online");  **Serial**.println("Updating time...");  configTime(gmtOffset\_sec, daylightOffset\_sec, ntpServer); // Configure the Time Service  timeClient.begin();  timeClient.update();  }  void loop() {  float h = dht.readHumidity();  float t = dht.readTemperature();  timeClient.update();  time\_t currentEpoch = timeClient.getEpochTime() + (gmtOffset\_sec);  tm \*currentTime = localtime(&currentEpoch);  display.clearDisplay();  display.setTextSize(1);  display.setTextColor(SSD1306\_WHITE);  display.setCursor(0,0);  display.print("Nama: ");  display.println(NAMA);  display.print("NIM: ");  display.println(NIM);  display.print("Kelas: ");  display.println(KELAS);  display.setCursor(0, 20);  display.print("Waktu WIB: ");  display.print(currentTime->tm\_hour);  display.print(":");  if (currentTime->tm\_min < 10) {  display.print("0");  }  display.print(currentTime->tm\_min);  display.print(":");  if (currentTime->tm\_sec < 10) {  display.print("0");  }  display.println(currentTime->tm\_sec);  display.setCursor(0, 40);  display.print("WiFi: ");  display.println(WiFi.status() == WL\_CONNECTED ? "Connected" : "Disconnected");  display.print("Date: ");  display.print(currentTime->tm\_year + 1900);  display.print("-");  display.print(currentTime->tm\_mon + 1);  display.print("-");  display.println(currentTime->tm\_mday);  display.display();  delay(2000); // Pause for 2 seconds  display.clearDisplay();  display.setTextSize(1);  display.setTextColor(SSD1306\_WHITE);  display.setCursor(0,0);  display.print("Temperature: ");  display.print(t);  display.println(" C");  display.print("            ");  display.print(t \* 9.0 / 5.0 + 32.0);  display.println(" F");  display.print("            ");  display.print(t + 273.15);  display.println(" K");  display.print("Humidity: ");  display.print(h);  display.println(" %");  display.display();  delay(2000); // Pause for 2 seconds  struct tm timeinfo;  if (!getLocalTime(&timeinfo)) {  **Serial**.println("Connection Err");  return;  }  **Serial**.println(&timeinfo, "%H:%M:%S");  **Serial**.println(&timeinfo, "%d/%m/%Y");  delay(250);  } |

1. Hasil Pengerjakan (screenshoot dijalankan)

|  |
| --- |
| Output 1 |
|  |

|  |
| --- |
| Output 2 |
|  |

|  |
| --- |
| Output 3 |
|  |