1. **NO. JOBSHEET** : 3
2. **JUDUL** : TOPOLOGI JARINGAN LOKAL DAN WIFI
3. **TUJUAN**
    1) Mahasiswa dapat memahami cara kerja protokol topologi jaringan lokal yang memanfaatkan Wi-Fi untuk berkomunikasi.
    2) Mahasiswa dapat merancang topologi jaringan yang memanfaatkan Wi-Fi untuk penerapan Wireless Sensor Network (WSN) dan Internet of Things (IoT).
    3) Mahasiswa dapat memilih dan menggunakan topologi jaringan yang tepat sesuai dengan kondisi lapangan untuk penerapan WSN dan IoT.
4. **ALAT DAN BAHAN**
    1) ESP32
    2) Breadboard
    3) Kabel jumper
    4) Sensor DHT 11, RFID
    5) LED (5) dan Push Button (3)
    6) Servo
    7) Resistor 330,1K, 10K Ohm (@ 3)
5. **TEORI SINGKAT**

Wireless Fidelity atau yang lebih awam kita sebut wifi adalah suatu teknologi yang menggunakan gelombang radio dalam rentang 2,4GHz sampai dengan 5GHz untuk menghubungkan perangkat seperti PC/laptop, smartphone, dan perangkat microcontroller seperti ESP32 dan ESP8266 ke jaringan lokal wireless untuk bisa mengakses internet. Untuk dapat melakukan akses internet tersebut,maka perangkat elektronik diatas perlu berada dalam satu titik akses atau hotspot jaringan nirkabel sehingga terhubung dengan wifi. Pada umumnya jaringan wifi dapat menjangkau hingga 20 meter didalam ruangan dan lebih dari 20 meter untuk di luar ruangan. Pada awal kemunculannya, wifi hanya digunakan sebagai perangkat nirkabel pada jaringan LAN (Local Area Network) akan tetapi karena pesatnya teknologi di zaman sekarang wifi menjadi kebutuhan sehari-hari untuk akses jaringan internet dan IoT.

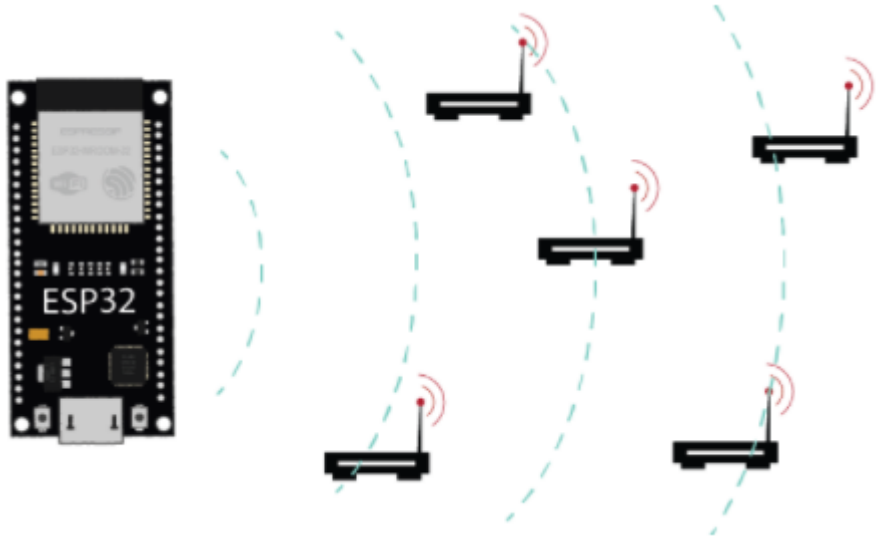
Berbagai data yang kita minta atau kirimkan melalui wifi didistribusikan melalui gelombang radio di udara. Supaya data tersebut bisa terbaca maka harus ada yang namanya wireless adaptor yang menghubungkan ke wifi. Gelombang radio yang berwujud sinyal ini lalu dikirim menuju router yang fungsinya untuk

memecahkan kode. Setelah terbaca maka data dikirim ke jaringan internet yang memanfaatkan koneksi ethernet. Karena jaringan wifi ini bekerja dua arah maka tiap data yang diterima dalam waktu yang sama menjadi kode pada tiap paket data lalu dikirim kembali dalam bentuk sinyal radio yang diterima adaptor komputer nirkabel.

6. LANGKAH PERCOBAAN

A. ESP32 Wi-Fi Modes dan Wifi-Scan

1. Pada ESP32, terdapat 3 mode akses untuk Wifi, yaitu WIFI_STA (station mode : ESP32 sebagai client yang terkoneksi ke access point), WIFI_AP (access point mode : ESP32 berperan sebagai access point), WIFI_STA_AP (access point and station : ESP32 dapat terkoneksi dengan access point yang lain).



2. Buka Arduino IDE dan upload script program berikut ke ESP32 untuk melakukan scan jaringan Wi-Fi.

```
/ #include "WiFi.h"

void setup() {
  Serial.begin(115200);

  // Set WiFi to station mode and disconnect from an AP if it was previously connected
  WiFi.mode(WIFI_STA);
  WiFi.disconnect();
  delay(100);

  Serial.println("Setup done");
}
```

```
void loop() {
 Serial.println("scan start");

 // WiFi.scanNetworks will return the number of networks found
 int n = WiFi.scanNetworks();
 Serial.println("scan done");
 if (n == 0) {
    Serial.println("no networks found");
 } else {
   Serial.print(n);
   Serial.println(" networks found");
   for (int i = 0; i < n; ++i) {
     // Print SSID and RSSI for each network found
     Serial.print(i + 1);
     Serial.print(": ");
     Serial.print(WiFi.SSID(i));
     Serial.print(" (");
     Serial.print(WiFi.RSSI(i));
     Serial.print(")");
     Serial.println((WiFi.encryptionType(i) == WIFI_AUTH_OPEN)?" ":"*");
     delay(10);
   }
 }
 Serial.println("");

 // Wait a bit before scanning again
 delay(5000);
}
```

3. Buka serial monitor dan dokumentasikan outputnya.
4. Buatlah flow chart program diatas.

B. Menghubungkan ESP32 dengan Jaringan Wi-Fi

1. Buatlah program seperti script dibawah ini, kemudian upload program tersebut ke ESP32.

```
#include <WiFi.h>


// Replace with your network credentials (STATION)
const char* ssid = "REPLACE_WITH_YOUR_SSID";
const char* password = "REPLACE_WITH_YOUR_PASSWORD";


void initWiFi() {
 WiFi.mode(WIFI_STA);
 WiFi.begin(ssid, password);
 Serial.print("Connecting to WiFi ..");
 while (WiFi.status() != WL_CONNECTED) {
  Serial.print('.');
  delay(1000);
```

```
  }
  Serial.println(WiFi.localIP());
}

void setup() {
  Serial.begin(115200);
  initWiFi();
  Serial.print("RRSI: ");
  Serial.println(WiFi.RSSI());
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

2. Buka serial monitor, kemudian dokumentasikan outputnya.
3. Buatlah flow chart program diatas.

C. Menghubungkan Kembali (Re-connect) ESP32 dengan Jaringan Wi-Fi

1. Buatlah program seperti script dibawah ini, kemudian upload program tersebut ke ESP32.

```
#include <WiFi.h>

// Replace with your network credentials (STATION)
const char* ssid = "REPLACE_WITH_YOUR_SSID";
const char* password = "REPLACE_WITH_YOUR_PASSWORD";

unsigned long previousMillis = 0;
unsigned long interval = 30000;

void initWiFi() {
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  Serial.print("Connecting to WiFi ..");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print('.');
    delay(1000);
  }
  Serial.println(WiFi.localIP());
}
```

```
void setup() {
  Serial.begin(115200);
  initWiFi();
  Serial.print("RRSI: ");
  Serial.println(WiFi.RSSI());
}

void loop() {
  unsigned long currentMillis = millis();
  // if WiFi is down, try reconnecting every CHECK_WIFI_TIME seconds
  if ((WiFi.status() != WL_CONNECTED) && (currentMillis - previousMillis >=interval)) {
    Serial.print(millis());
    Serial.println("Reconnecting to WiFi...");
    WiFi.disconnect();
    WiFi.reconnect();
    previousMillis = currentMillis;
  }
}
```

2. Buka serial monitor, kemudian matikan paket data sebentar hingga koneksi ESP32 dengan jaringna Wi-Fi terputus. Setelah itu, nyalakan lagi paket data. Dokumentasikan proses yang terjadi.
3. Buatlah flow chart program diatas.

D. Mengganti Hostname ESP32

1. Buatlah program seperti script dibawah ini, kemudian upload program tersebut ke ESP32.

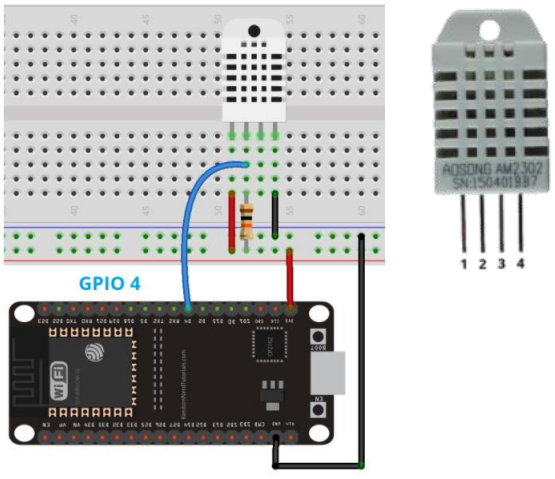
```
#include <WiFi.h>

// Replace with your network credentials (STATION)
const char* ssid = "REPLACE_WITH_YOUR_SSID";
const char* password = "REPLACE_WITH_YOUR_PASSWORD";

String hostname = "ESP32 Node Temperature";

void initWiFi() {
  WiFi.mode(WIFI_STA);
  WiFi.config(INADDR_NONE, INADDR_NONE, INADDR_NONE, INADDR_NONE);
```

```
  WiFi.setHostname(hostname.c_str()); //define hostname
  //wifi_station_set_hostname( hostname.c_str() );
  WiFi.begin(ssid, password);
  Serial.print("Connecting to WiFi ..");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print('.');
    delay(1000);
  }
  Serial.println(WiFi.localIP());
}

void setup() {
  Serial.begin(115200);
  initWiFi();
  Serial.print("RRSI: ");
  Serial.println(WiFi.RSSI());
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

2. Buka serial monitor, kemudian aktifkan mode koneksi Wi-Fi pada Smart-Phone atau Laptop. Lakukan scan dan lihat daftar jaringan Wi-Fi yang tersedia. Dokumentasikan hasil keluarannya.
3. Buatlah flow chart program diatas.

E. Mengirim Data Sensor ke Database

1. Buatlah rangkaian seperti Gambar di bawah ini.



1. Install library Asynch Web Server dan Asycnh TCP untuk ESP 32 dengan cara download dari link berikut ini.
   a. https://github.com/me-no-dev/ESPAsyncWebServer
   b. https://github.com/me-no-dev/AsyncTCP/archive/master.zip

   Install library tersebut secara manual dengan cara menyalin folder hasil ekstraksi file.zip ke direktori libarary Arduino di folder Document.
2. Buatlah script program seperti berikut ini.

```
// Import required libraries
#include "WiFi.h"
#include "ESPAsyncWebServer.h"
#include <Adafruit_Sensor.h>
#include <DHT.h>

// Replace with your network credentials
const char* ssid = "REPLACE_WITH_YOUR_SSID";
const char* password = "REPLACE_WITH_YOUR_PASSWORD";

#define DHTPIN 4     // Digital pin connected to the DHT sensor

// Uncomment the type of sensor in use:
#define DHTTYPE    DHT11     // DHT 11

DHT dht(DHTPIN, DHTTYPE);

// Create AsyncWebServer object on port 80
AsyncWebServer server(80);

String readDHTTemperature() {
 // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
 // Read temperature as Celsius (the default)
 float t = dht.readTemperature();
 // Read temperature as Fahrenheit (isFahrenheit = true)
 //float t = dht.readTemperature(true);
 // Check if any reads failed and exit early (to try again).
 if (isnan(t)) {
  Serial.println("Failed to read from DHT sensor!");
  return "--";
 }
 else {
  Serial.println(t);
  return String(t);
 }
}

String readDHTHumidity() {
 // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
 float h = dht.readHumidity();
 if (isnan(h)) {
  Serial.println("Failed to read from DHT sensor!");
```

```
   return "--";
  }
  else {
   Serial.println(h);
   return String(h);
  }
}

const char index_html[] PROGMEM = R"rawliteral(
<!DOCTYPE HTML><html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.2/css/all.css"
integrity="sha384-
fnmOCqbTlWIlj8LyTjo7mOUStjsKC4pOpQbqyi7RrhN7udi9RwhKkMHpvLbHG9Sr"
crossorigin="anonymous">
  <style>
    html {
     font-family: Arial;
     display: inline-block;
     margin: 0px auto;
     text-align: center;
    }
    h2 { font-size: 3.0rem; }
    p { font-size: 3.0rem; }
    .units { font-size: 1.2rem; }
    .dht-labels{
      font-size: 1.5rem;
      vertical-align:middle;
      padding-bottom: 15px;
    }
  </style>
</head>
<body>
  <h2>ESP32 DHT Server</h2>
  <p>
    <i class="fas fa-thermometer-half" style="color:#059e8a;"></i>
    <span class="dht-labels">Temperature</span>
    <span id="temperature">%TEMPERATURE%</span>
    <sup class="units">&deg;C</sup>
  </p>
  <p>
    <i class="fas fa-tint" style="color:#00add6;"></i>
    <span class="dht-labels">Humidity</span>
    <span id="humidity">%HUMIDITY%</span>
    <sup class="units">&percnt;</sup>
  </p>
</body>
<script>
setInterval(function ( ) {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("temperature").innerHTML = this.responseText;
    }
```

```
  };
  xhttp.open("GET", "/temperature", true);
  xhttp.send();
}, 10000 ) ;

setInterval(function ( ) {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("humidity").innerHTML = this.responseText;
    }
  };
  xhttp.open("GET", "/humidity", true);
  xhttp.send();
}, 10000 ) ;
</script>
</html>)rawliteral";

// Replaces placeholder with DHT values
String processor(const String& var){
  //Serial.println(var);
  if(var == "TEMPERATURE"){
    return readDHTTemperature();
  }
  else if(var == "HUMIDITY"){
    return readDHTHumidity();
  }
  return String();
}

void setup(){
  // Serial port for debugging purposes
  Serial.begin(115200);

  dht.begin();

  // Connect to Wi-Fi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi..");
  }

  // Print ESP32 Local IP Address
  Serial.println(WiFi.localIP());

  // Route for root / web page
  server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send_P(200, "text/html", index_html, processor);
  });
  server.on("/temperature", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send_P(200, "text/plain", readDHTTemperature().c_str());
  });
  server.on("/humidity", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send_P(200, "text/plain", readDHTHumidity().c_str());
```

```
  });

  // Start server
  server.begin();
}

void loop(){

}
```

3. Upload program di atas. Kemudian buka serial monitor untuk mengetahui IP Address ESP32.
4. Akses IP Address ESP32 pada browser laptop. Dokumentasikan hasilnya dan buatlah flow chart dari program tersebut.

7. PERTANYAAN DAN TUGAS

1. Buatlah program pada ESP32 dengan urutan proses seperti berikut ini, agar ESP32 bisa melakukan setting SSID dan Passsword lebih fleksibel.
    a. Ketika pertama kali booting, mode ESP32 adalah Station Mode untuk membaca kombinasi SSID dan Password pada jaringan sebelumnya.
    b. Jika gagal, ESP32 akan berubah mode manjadi Access Point Mode dan membuat jaringan Wi-Fi tanpa proteksi/tanpa password, agar user dapat terhubung dengan ESP32.
    c. Setelah itu, hubungkan laptop dengan ESP32 dan akses IP Address ESP32 (ESP32 Web Server) pada browser laptop untuk membua konfigurasi SSID dan Password dan menyimpannya pada EEPROM.
    d. Kemudian matikan ESP32. ESP32 akan berusaha terhubung dengan jaringan yang telah dikonfigurasi sebelumnya.
    e. Jika berhasil terhubung, pada serial monitor akan terdapat pesan **Connected to “SSID” Successfully**.
    f. Jika gagal terhubung, ESP32 akan masuk pada mode Access Point kembali untuk melakukan konfigurasi SSID dan Password.