

MODUL 2

KONEKSI INTERNET MELALUI WIFI



CAPAIAN PEMBELAJARAN

Mampu mengkoneksikan sistem Embedded ke jaringan internet melalui Wifi

KEBUTUHAN ALAT/BAHAN/SOFTWARE

Hardware:

1. Modul ESP32 DEV
2. LED x1
3. Resistor x1

Software:

1. Arduino IDE (<http://arduino.cc>)

DASAR TEORI

Termasuk dalam komponen *Internet of Things* adalah koneksi Internet. Koneksi Internet dapat dilakukan dengan menggunakan jaringan kabel maupun nirkabel. Koneksi jaringan kabel menggunakan protokol Ethernet sedangkan koneksi nirkabel menggunakan jaringan Wifi.

Koneksi Internet menggunakan Wifi membutuhkan komponen Access Point dan interface Wifi. ESP32 sudah dilengkapi dengan interface Wifi yang dapat dikonfigurasi sebagai client Wifi (STA), Access Point (AP), dan STA dan AP.

Agar dapat berfungsi interface Wifi membutuhkan program yang dapat dibuat sendiri atau menggunakan hasil kerja orang lain yang dikemas dalam pustaka fungsi (library). Library yang digunakan untuk Wifi ESP32 adalah Wifi.h.

Tabel 1 Status Wifi

Nilai	Konstan (status)	Arti
0	WL_IDLE_STATU	temporary status assigned when WiFi.begin() is called
1	WL_NO_SSID_AVAIL	when no SSID are available
2	WL_SCAN_COMPLETED	scan networks is completed
3	WL_CONNECTED	when connected to a WiFi network
4	L_CONNECT_FAILED	when the connection fails for all the attempts
5	WL_CONNECTION_LOST	(when the connection is lost)
6	WL_DISCONNECTED	when disconnected from a network

PRAKTIK

A. Scan Access Point Yang Ada

1. Tulis program berikut ini, upload dan lihat hasilnya

```
#include "WiFi.h"
void setup()
{
    Serial.begin(115200);
    WiFi.mode(WIFI_STA); //station
    WiFi.disconnect(); // disconnected
    delay(100);
    Serial.println("Setup selesai");
}

void loop()
{
    Serial.println("Scan dimulai");
```

```

int n = WiFi.scanNetworks(); //mulai scan
Serial.println("Scan selesai");
if (n == 0) {
    Serial.println("Tidak ada Wifi");
} else {
    Serial.print(n);
    Serial.println(" jaringan terciduk");
    Serial.println("Nr | SSID
| RSSI | CH | Encryption");
    for (int i = 0; i < n; ++i) {
        Serial.printf("%2d", i + 1);
        Serial.print(" | ");
        Serial.printf("%-32.32s",
WiFi.SSID(i).c_str());
        Serial.print(" | ");
        Serial.printf("%4d", WiFi.RSSI(i));
        Serial.print(" | ");
        Serial.printf("%2d", WiFi.channel(i));
        Serial.print(" | ");
        switch (WiFi.encryptionType(i))
        {
            case WIFI_AUTH_OPEN:
                Serial.print("open");
                break;
            case WIFI_AUTH_WEP:
                Serial.print("WEP");
                break;
            case WIFI_AUTH_WPA_PSK:
                Serial.print("WPA");
                break;
            case WIFI_AUTH_WPA2_PSK:
                Serial.print("WPA2");
                break;
            case WIFI_AUTH_WPA_WPA2_PSK:
                Serial.print("WPA+WPA2");
                break;
            case WIFI_AUTH_WPA2_ENTERPRISE:
                Serial.print("WPA2-EAP");
                break;
            case WIFI_AUTH_WPA3_PSK:
                Serial.print("WPA3");

```

```

        break;
    case WIFI_AUTH_WPA2_WPA3_PSK:
        Serial.print("WPA2+WPA3");
        break;
    case WIFI_AUTH_WAPI_PSK:
        Serial.print("WAPI");
        break;
    default:
        Serial.print("unknown");
    }
    Serial.println();
    delay(10);
}
}
Serial.println("");
WiFi.scanDelete();

delay(5000);

```

2. Catat access point yang paling kuat signalnya!

B. Koneksi ke Access Point

```

#include "WiFi.h"
#define LEDStatus 12
const char* ssid      = "harjolukito";
const char* password = "amil971da";

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.print("IP Address: ");
    Serial.println(WiFi.localIP());
}

void setup()
{

```

```

        Serial.begin(115200);
        pinMode(LEDStatus,HIGH);
        initWiFi();
    }

    void loop()
    {

    }

```

C. Konfigurasi Interface Secara Statis

1. Tulis program berikut upload lihat hasilnya

```

#include "WiFi.h"

const char* ssid      = "SSID";
const char* password = "password";

//Set IPAddress sesuaikan dengan jaringan lokal
IPAddress local_IP(10, 10, 65, 235);
IPAddress gateway(10, 10, 645, 1);
IPAddress subnet(255, 255, 240, 0);
IPAddress primaryDNS(8, 8, 8, 8);

void setup()
{
    Serial.begin(115200);
    WiFi.mode(WIFI_STA);//station
    Serial.print("Connecting to ");
    Serial.println(ssid);
    WiFi.disconnect();
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.println("WiFi connected.");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
}

```

```
void loop()  
{  
  
}
```

C. Pengecekan Versi Hardware Ethernet

1. Tulis program berikut ini, upload dan jalankan



LATIHAN

1. Buat program untuk menentukan *Access Point* yang paling kuat sinyalnya.
2. Buat program (atau menambahkan Latihan 1) agar melakukan koneksi ulang jika terputus.
3. Tambahkan kode program ke latihan 2 untuk indikator status Wifi (terutama connect dan disconnect)

TUGAS

1. Buat diagram alir untuk hasil latihan.

REFERENSI

1. Neil Kolban, Kolban's Book on ESP8266, 2016