

LAMPIRAN

1. Node Transmitter

```
#include <WiFi.h>
#include <ThingESP.h>
#include <SPI.h>
#include <LoRa.h>

ThingESP32 thing("jalal", "LORAWA", "123123123");

#define SS 5
#define RST 14
#define DIO0 26

bool tungguBalasan = false;

// ===== HANDLE WA =====
String HandleResponse(String query) {

    query.trim();
    query.toLowerCase();

    if (query == "p") {
        return "MENU:\n1. Hidupkan Lampu\n2. Matikan Lampu";
    }

    else if (query == "1") {

        Serial.println("Kirim: ON");

        LoRa.beginPacket();
        LoRa.print("ON");
        LoRa.endPacket();

        tungguBalasan = true;

        return "Perintah dikirim: Lampu HIDUP";
    }

    else if (query == "2") {
```

```

        Serial.println("Kirim: OFF");

        LoRa.beginPacket();
        LoRa.print("OFF");
        LoRa.endPacket();

        tungguBalasan = true;

        return "Perintah dikirim: Lampu MATI";
    }

    else {
        return "Perintah tidak dikenal";
    }
}

void setup() {
    Serial.begin(115200);
    delay(1000);

    Serial.println("=== STARTING TX ===");

    // WiFi
    thing.SetWiFi("GRATIS", "aman1234");
    thing.initDevice();
    thing.setCallback(HandleResponse);

    // LoRa
    SPI.begin(18, 19, 23, SS);
    LoRa.setPins(SS, RST, DIO0);

    int retry = 0;
    while (!LoRa.begin(433E6)) {
        Serial.println("LoRa gagal...");
        retry++;

        if (retry >= 5) {
            ESP.restart();
        }

        delay(2000);
    }

    LoRa.setTxPower(17);
    LoRa.setSpreadingFactor(7);

```

```

    LoRa.setSignalBandwidth(125E3);

    Serial.println("TX Siap");
}

void loop() {
    thing.Handle();
    yield();

    int packetSize = LoRa.parsePacket();

    if (packetSize && tungguBalasan) {
        String pesan = "";

        while (LoRa.available()) {
            pesan += (char)LoRa.read();
        }

        pesan.trim();

        Serial.println("Dari RX: " + pesan);

        // 🌟 Kirim pesan kedua (STATUS)
        String balasan = "Status: " + pesan;

        thing.sendMessage("628XXXXXXXXXX", balasan);

        tungguBalasan = false;
    }
}

```

2. Node Receiver

```

#include <SPI.h>
#include <LoRa.h>

#define SS 5
#define RST 14
#define DI00 26

#define RELAY 27

```

```

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

  pinMode(RELAY, OUTPUT);
  digitalWrite(RELAY, HIGH);

  SPI.begin(18, 19, 23, SS);
  LoRa.setPins(SS, RST, DI00);

  while (!LoRa.begin(433E6)) {
    Serial.println("LoRa gagal...");
    delay(2000);
  }

  LoRa.setTxPower(17);
  LoRa.setSpreadingFactor(7);
  LoRa.setSignalBandwidth(125E3);

  Serial.println("RX Siap");
}

void loop() {
  int packetSize = LoRa.parsePacket();

  if (packetSize) {
    String pesan = "";

    while (LoRa.available()) {
      pesan += (char)LoRa.read();
    }

    pesan.trim();

    Serial.println("Terima: " + pesan);

    if (pesan == "ON") {
      digitalWrite(RELAY, LOW);
      kirimStatus("LAMPU HIDUP");
    }

    else if (pesan == "OFF") {
      digitalWrite(RELAY, HIGH);
      kirimStatus("LAMPU MATI");
    }
  }
}

```

```
    }  
  }  
  
  void kirimStatus(String status) {  
    LoRa.beginPacket();  
    LoRa.print(status);  
    LoRa.endPacket();  
  
    Serial.println("Kirim balik: " + status);  
  }
```