

LAMPIRAN I

Source Code Program

```
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
RF24 radio(9, 10);
// =====
// KONFIGURASI NODE (UNTUK 5 LAMPU)
// =====
// Ubah NODE_ID sesuai alat: 1, 2, 3, 4, atau 5
#define NODE_ID 1
// =====
// DEFINISI PIN
// =====
const int pirPin      = 2;    // Pin Sensor Gerak
PIR
const int relayPin     = 8;    // Pin Relay / Lampu
const int pinDeteksiAC = 4;    // Pin Deteksi
Listrik AC
// =====
// KONFIGURASI WAKTU
// =====
unsigned long lampTimer = 0;
const unsigned long lampDuration = 15000; // Lampu
nyala 15 detik (bisa diubah)
// =====
// VARIABEL STATUS
// =====
bool lampState = false;
bool localTrigger = false;
```

```

bool adaListrikAC = false; // Status deteksi
listrik AC
// =====
// STRUKTUR DATA KOMUNIKASI
// =====
struct Data {
    int sender;
};
Data data;
// =====
// ALAMAT PIPA KOMUNIKASI (CUKUP 5 ALAMAT)
// =====
const byte address1[6] = "00001";
const byte address2[6] = "00002";
const byte address3[6] = "00003";
const byte address4[6] = "00004";
const byte address5[6] = "00005";
const byte* getAddress(int id) {
    switch(id) {
        case 1: return address1;
        case 2: return address2;
        case 3: return address3;
        case 4: return address4;
        case 5: return address5;
        default: return address1;
    }
}
// =====
// SETUP AWAL
// =====
void setup() {
    pinMode(pirPin, INPUT);
    pinMode(relayPin, OUTPUT);
}

```

```

    pinMode(pinDeteksiAC, INPUT); // Set pin deteksi
    sebagai INPUT
    digitalWrite(relayPin, LOW); // Awal lampu mati

    Serial.begin(9600);
    Serial.println("SISTEM LAMPU DARURAT 5 NODE");
    Serial.println("Tunggu sensor siap...");
    delay(30000); // Tunggu PIR stabil
    Serial.println("Siap bekerja.");

    // Inisialisasi Modul NRF24L01
    if (!radio.begin()) {
        Serial.println("RADIO GAGAL INISIALISASI!");
        while(1);
    }
    radio.setAutoAck(true);
    radio.enableDynamicPayloads();
    radio.setRetries(5, 15);
    radio.setChannel(108); // WAJIB SAMA di
    semua node
    radio.setPALevel(RF24_PA_MIN);
    radio.setDataRate(RF24_250KBPS);
    radio.openReadingPipe(0, getAddress(NODE_ID));
    radio.startListening();

    Serial.print("NODE KE-");
    Serial.println(NODE_ID);
    Serial.println("-----");
}
// =====
// LOOP UTAMA
// =====
void loop() {

```

```

unsigned long now = millis();
// 1. BACA STATUS LISTRIK AC
cekStatusAC();
// 2. KEPUTUSAN SISTEM BERDASARKAN LISTRIK
if (adaListrikAC == true) {
    // == KONDISI: ADA LISTRIK AC ==
    matikanSistemTotal();
} else {
    // == KONDISI: MATI LISTRIK / PAKAI SISTEM
    SUMBER ENERGI LISTRIK CADANGAN ==
    jalankanSistemDarurat(now);
}
}
// =====
// FUNGSI CEK DETEKSI AC
// =====
void cekStatusAC() {
    // LOGIKA:
    // LOW = Ada Listrik AC
    // HIGH = Tidak Ada Listrik / Pakai SISTEM SUMBER
    ENERGI LISTRIK CADANGAN
    if (digitalRead(pinDeteksiAC) == LOW) {
        adaListrikAC = true;
    } else {
        adaListrikAC = false;
    }
}
// =====
// FUNGSI SAAT ADA LISTRIK AC
// =====
void matikanSistemTotal() {
    digitalWrite(relayPin, LOW); // Pastikan lampu
    mati

```

```

    lampState = false;                // Reset status
lampu
    localTrigger = false;            // Reset sinyal
 kirim
    // Cetak status agar serial monitor tidak penuh
    static unsigned long lastPrint = 0;
    if (millis() - lastPrint > 1000) {
        Serial.println("STATUS: ADA LISTRIK AC ->
SISTEM NONAKTIF");
        lastPrint = millis();
    }
}
// =====
// FUNGSI SAAT TIDAK ADA LISTRIK / PAKAI SISTEM
SUMBER ENERGI LISTRIK CADANGAN
// =====
void jalankanSistemDarurat(unsigned long now) {
    static unsigned long lastPrint = 0;
    if (millis() - lastPrint > 1000) {
        Serial.println("STATUS: MATI LISTRIK -> SISTEM
AKTIF (Mode Darurat)");
        Serial.println("Lampu TETAP MATI sampai ada
gerakan terdeteksi");
        lastPrint = millis();
    }
    // Lampu hanya nyala jika ada pemicu:
    checkPIR(now);                // Baca sensor gerak ->
pemicu utama
    receiveSignal(now);           // Terima sinyal dari node
lain -> pemicu kedua
    updateLamp(now);              // Atur waktu nyala/mati
lampu
}

```

```

// =====
// CEK SENSOR GERAK LOKAL
// =====
void checkPIR(unsigned long now) {
    bool pirValue = digitalRead(pirPin);
    if (pirValue == HIGH) {
        // FITUR BARU: Reset timer setiap kali ada
gerakan terdeteksi
        lampTimer = now;
        if (!lampState) {
            lampState = true;
            digitalWrite(relayPin, HIGH);
            Serial.print("NODE ");
            Serial.print(NODE_ID);
            Serial.println(": GERAK TERDETEKSI -> LAMPU
NYALA");
        } else {
            Serial.println("Gerakan terdeteksi ULANG ->
Timer di-reset ulang jadi 15 detik");
        }
        // Kirim sinyal ke node tetangga
        if (!localTrigger) {
            localTrigger = true;
            Serial.println("Mengirim sinyal ke node
lain...");
            sendSignal();
        }
    } else {
        localTrigger = false;
    }
}
// =====
// KIRIM SINYAL KE NODE LAIN (KHUSUS 5 NODE)

```

```
// =====
void sendSignal() {
    // Kirim ke Node KANAN
    if (NODE_ID < 5) { // Node 1,2,3,4 kirim ke kanan
        radio.stopListening();
        delay(10);
        radio.openWritingPipe(getAddress(NODE_ID + 1));
        data.sender = NODE_ID;
        bool ok = radio.write(&data, sizeof(data));
        if (ok) {
            Serial.print("Kirim ke Node ");
            Serial.print(NODE_ID + 1);
            Serial.println(" BERHASIL");
        } else {
            Serial.print("Kirim ke Node ");
            Serial.print(NODE_ID + 1);
            Serial.println(" GAGAL");
        }
        delay(10);
        radio.startListening();
    }
    // Kirim ke Node KIRI
    if (NODE_ID > 1) { // Node 2,3,4,5 kirim ke kiri
        radio.stopListening();
        delay(10);
        radio.openWritingPipe(getAddress(NODE_ID - 1));
        data.sender = NODE_ID;
        bool ok = radio.write(&data, sizeof(data));
        if (ok) {
            Serial.print("Kirim ke Node ");
            Serial.print(NODE_ID - 1);
            Serial.println(" BERHASIL");
        } else {
```

```

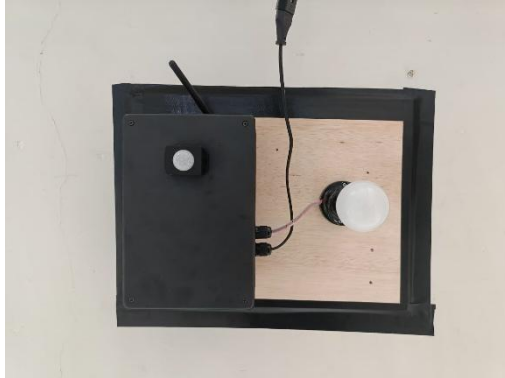
        Serial.print("Kirim ke Node ");
        Serial.print(NODE_ID - 1);
        Serial.println(" GAGAL");
    }
    delay(10);
    radio.startListening();
}
}
// =====
// TERIMA SINYAL DARI LUAR
// =====
void receiveSignal(unsigned long now) {
    if (radio.available()) {
        radio.read(&data, sizeof(data));
        Serial.print("Menerima sinyal DARI NODE ");
        Serial.println(data.sender);
        // FITUR BARU: Reset timer juga kalau dapat
sinyal dari node lain
        lampTimer = now;
        // Nyalakan lampu hanya saat dapat sinyal
        if (!lampState) {
            lampState = true;
            digitalWrite(relayPin, HIGH);
            Serial.println("-> LAMPU NYALA karena sinyal
tetangga");
        } else {
            Serial.println("-> Dapat sinyal ulang ->
Timer di-reset ulang jadi 15 detik");
        }
    }
}
}
// =====
// ATUR WAKTU MATI LAMPU

```

```
// =====  
void updateLamp(unsigned long now) {  
    if (lampState) {  
        if (now - lampTimer >= lampDuration) {  
            lampState = false;  
            digitalWrite(relayPin, LOW);  
            Serial.println("Lampu MATI (Habis Waktu 15  
detik tanpa gerakan)");  
        }  
    }  
}
```

LAMPIRAN II

REALISASI ALAT



Gambar 6.1 Alat Tampak Depan