DAFTAR PUSTAKA

Abinaya, R., Varsha, V. and Hariharan, K. (2017) 'An intelligent street light system based on piezoelectric sensor networks', in *4th International Conference on Electronics and Communication Systems (ICECS)*, pp. 1–7. Available at: https://doi.org/https://doi.org/10.1109/ECS.2017.8067855.

Adelantado, F. *et al.* (2017) 'Understanding the Limits of LoRaWAN', *IEEE Communications Magazine*, 55(9), pp. 34–40. Available at: https://doi.org/10.1109/MCOM.2017.1600613.

Al-Smadi, A.M. *et al.* (2019) 'Street Lighting Energy-Saving System', in *2019 16th International Multi-Conference on Systems, Signals & Devices (SSD)*, pp. 763–766. Available at: https://doi.org/10.1109/SSD.2019.8893160.

Badarov, D.H. and Manoev, V.D. (2017) 'Teaching methodology for digital pulse width modulation and phase angle control', in *XXVI International Scientific Conference Electronics (ET)*. Available at: https://doi.org/https://doi.org/10.1109/ET.2017.8124335.

B, K. *et al.* (2021) 'Efficient Power Generation to Automated Street Lights based on Traffic Density', in *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 1006–1010. Available at: https://doi.org/10.1109/ICCMC51019.2021.9418389.

Campisi, D., Gitto, S. and Morea, D. (2018) 'Economic feasibility of energy efficiency improvements in street lighting systems in Rome', *Journal of Cleaner Production*, 175, pp. 190–198. Available at: https://doi.org/10.1016/j.jclepro.2017.12.063.

Daely, P.T. *et al.* (2017) 'Design of smart LED streetlight system for smart city with web-based management system', *IEEE Sensors Journal*, 17(18), pp. 6100–6110. Available at: https://doi.org/10.1109/JSEN.2017.2734101.

Direktorat Jenderal Pengendalian Perubahan Iklim - Kementerian Lingkungan Hidup dan Kehutanan (2021) *UPDATED NATIONALLY DETERMINED CONTRIBUTION REPUBLIC OF INDONESIA* . Available at: https://ditjenppi.menlhk.go.id/reddplus/images/adminppi/adaptasi/dokumen/Updated_NDC.pdf (Accessed: 20 May 2023).

El-Zein, N. (2013) 'The LED Lighting Revolution', in *Sustainability, Energy and Architecture: Case Studies in Realizing Green Buildings*. Elsevier Inc., pp. 171–194. Available at: https://doi.org/10.1016/B978-0-12-397269-9.00007-4.

nicx Team (2021) *PWM : Pulse Width Modulation – explained*. Available at: https://www.gadgetronicx.com/pwm-pulse-width-modulation-explained/ (Accessed: 7 August 2023).

Gillis, A.S. (2021) *Network Topology*, *TechTarget*. Available at: https://www.techtarget.com/searchnetworking/definition/network-topology#:~:text=What%20is%20a%20network%20topology,often%20represented%20as%20a%20graph. (Accessed: 7 August 2023).

Jia, G. *et al.* (2018) 'SSL: Smart Street Lamp Based on Fog Computing for Smarter Cities', *IEEE Transactions on Industrial Informatics*, 14(11), pp. 4995–5004. Available at: https://doi.org/10.1109/TII.2018.2857918.

Jiang, Y. *et al.* (2021) 'An Energy-Efficient Street Lighting Approach Based on Traffic Parameters Measured by Wireless Sensing Technology', *IEEE Sensors Journal*, 21(17), pp. 19134–19143. Available at: https://doi.org/10.1109/JSEN.2021.3089208.

Kementerian Perhubungan (2018) *Peraturan Menteri Perhubungan Republik Indonesia Nomor PM 27 Tahun 2018 Tentang Alat Penerangan Jalan*. Indonesia: https://jdih.dephub.go.id/peraturan/detail?data=8ppVY1RjEjLJUAxd7XuLtJ4OZqhvNLXhC8LQySXMskWm4DuVjmsLLYm49XnCzhXDi18hixdyjR1xk8hdm475xnT24eRhkImxkln8MS9nuR95LG7PBdEfFm2yb0Mnn25i3ie0HmwMQewSCt7LrcoSXhlnLK.

Lelisa Army, W. *et al.* (2022) *TEKNOLOGI JARINGAN KOMPUTER*. Bandung: Widina Bhakti Persada Bandung . Available at: www.penerbitwidina.com.

LoRa Alliance® (2015) *A technical overview of LoRa ® and LoRaWAN ™ What is it?* Available at: https://lora-alliance.org/resource_hub/what-is-lorawan/ (Accessed: 29 May 2023).

Made Karmiathi, N., Gusti Putu Mastawan Eka Putra, I. and Wayan Wisswani, N. (2019) 'Analisis Teknis-Ekonomis Retrofit Lampu Penerangan Jalan Raya Gatot Subroto-Denpasar', *JURNAL MATRIX*, 9(3), p. 83.

Modabbir and Mohammad, A. (2021) 'Energy and Economic Analysis of Smart Technologies on Street Lighting System', in *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pp. 389–392. Available at: https://doi.org/10.1109/ICACCS51430.2021.9441734.

Nasir, B., Abdullah, Z. and Yaseen, M. (2019) 'Efficiency Calculation of LEDs Used In Street Lighting Lamps By Matlab', in *2019 International Conference on Computing and Information Science and Technology and Their Applications (ICCISTA)*, pp. 1–5. Available at: https://doi.org/10.1109/ICCISTA.2019.8830660.

ndall *et al.* (2021) 'Monitoring for Analytes through LoRa and LoRaWAN Technology', in *Procedia Computer Science*. Elsevier B.V., pp. 152–159. Available at: https://doi.org/10.1016/j.procs.2021.05.041.

Pereira, A.M.E. *et al.* (2015) 'Some considerations about LED technology in public lighting', in *2015 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*, pp. 561–565. Available at: https://doi.org/10.1109/Chilecon.2015.7400433.

Putranto, H., Wibawanto, S. and Pradana, D.A. (2021) *Modul Perencanaan Pencahayaan Instalasi Penerangan Listrik*. Ahlimedia Press.

Sanchez-Sutil, F. and Cano-Ortega, A. (2021) 'Smart regulation and efficiency energy system for street lighting with LoRa LPWAN', *Sustainable Cities and Society*, 70, p. 102912. Available at: https://doi.org/https://doi.org/10.1016/j.scs.2021.102912.

Sekretariat Perusahaan PT PLN (Persero) (2021) *STATISTIK PLN 2021*. Jakarta. Available at: https://web.pln.co.id/stakeholder/laporan-statistik (Accessed: 21 May 2023).

Smith, C.B. and Parmenter, K.E. (2016) 'Lighting Management', in *Energy Management Principles*. Elsevier, pp. 189–218. Available at: https://doi.org/10.1016/b978-0-12-802506-2.00009-4.

Song, Y. *et al.* (2017) 'An Internet of Energy Things Based on Wireless LPWAN', *Engineering*, 3(4), pp. 460–466. Available at: https://doi.org/10.1016/J.ENG.2017.04.011.

Suhendra, T. *et al.* (2018) 'Kendali Kecepatan Motor DC dengan Metode Pulse Width Modulation menggunakan N-channel Mosfet', *Jurnal Sustainable: Jurnal Hasil Penelitian dan Industri Terapan* , 07(02), pp. 78–85. Available at: https://media.neliti.com/media/publications/266846-kendali-kecepatan-motor-dc-dengan-metode-a7110155.pdf (Accessed: 24 June 2023).

Syed, A. *et al.* (2004) 'Digital Pulse Width Modulator Architectures', in *351h Annual IEEE Power Electronics Specialirls Conference*. Available at: https://doi.org/https://doi.org/10.1109/PESC.2004.1354828.

Tambunan, J.M. *et al.* (2010) *Pencahayaan*. Institut Teknologi PLN. Available at: https://repo.itpln.ac.id/124/1/Pencahayaan.pdf (Accessed: 5 August 2023).

Thungtong, A., Chaichan, C. and Suwannarat, K. (2021) 'A web-based control system for traditional street lighting that uses high-pressure sodium lamps', *Heliyon*, 7(11), p. e08329. Available at: https://doi.org/https://doi.org/10.1016/j.heliyon.2021.e08329.

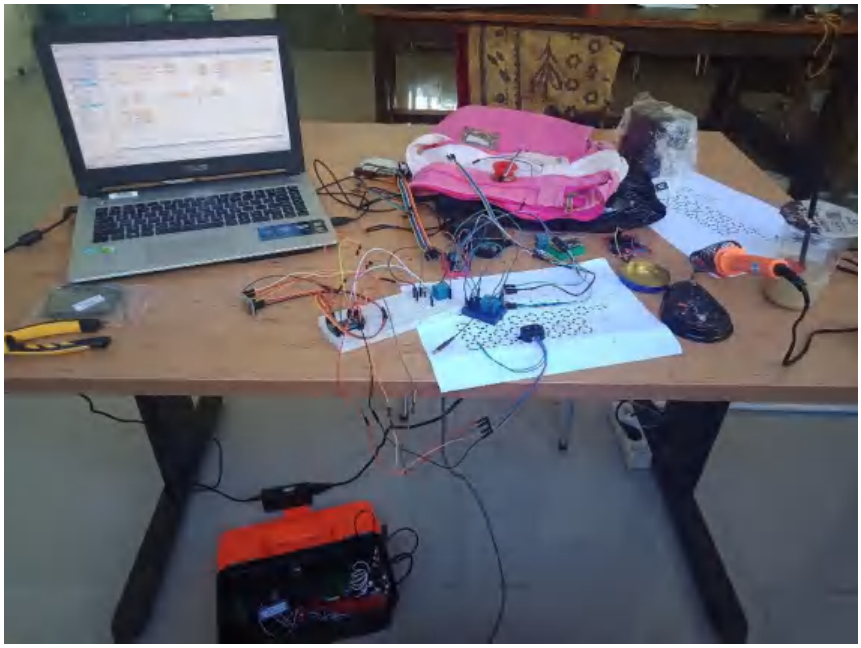
Xu, X., Zhan, A. and Li, X. (2019) 'Design and implementation of street light control system based on power line carrier communication', in *Procedia Computer Science*. Elsevier B.V., pp. 734–739. Available at: https://doi.org/10.1016/j.procs.2019.08.106.

Yang, Y.-S. *et al.* (2020) 'An Implementation of High Efficient Smart Street Light Management System for Smart City', *IEEE Access*, 8, pp. 38568–38585. Available at: https://doi.org/10.1109/ACCESS.2020.2975708.

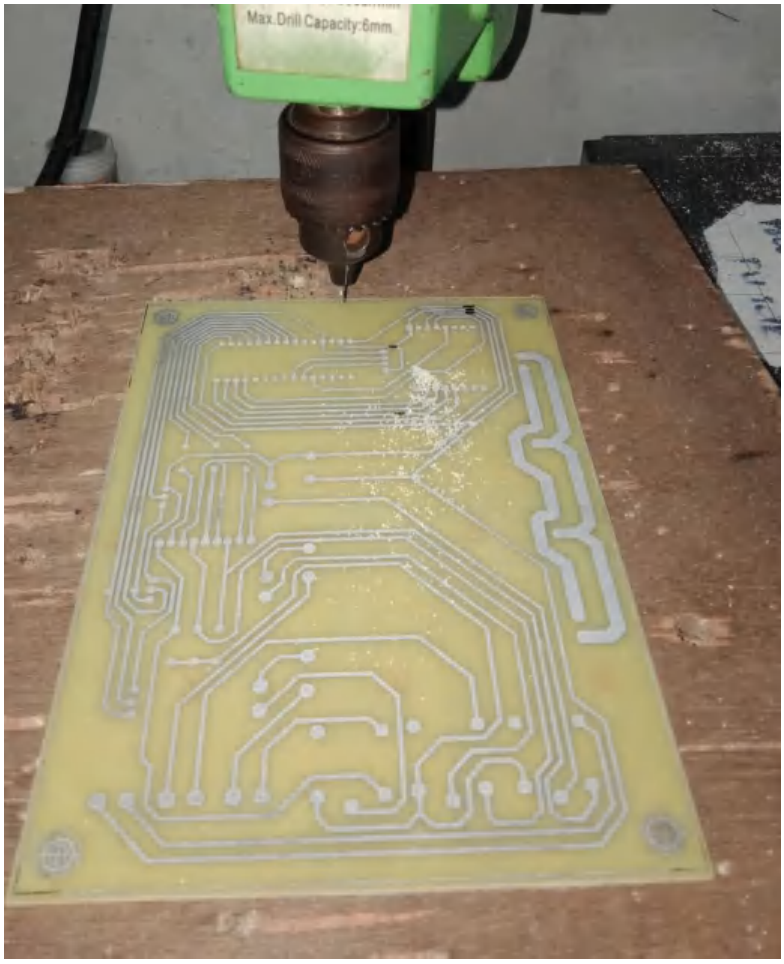
Ye, T. *et al.* (2017) 'The Study of Natural Exposure Testing for LED Lighting System', in *18th International Conference on Electronic Packaging Technology (ICEPT)*, pp. 726–730. Available at: https://doi.org/https://doi.org/10.1109/ICEPT.2017.8046552.

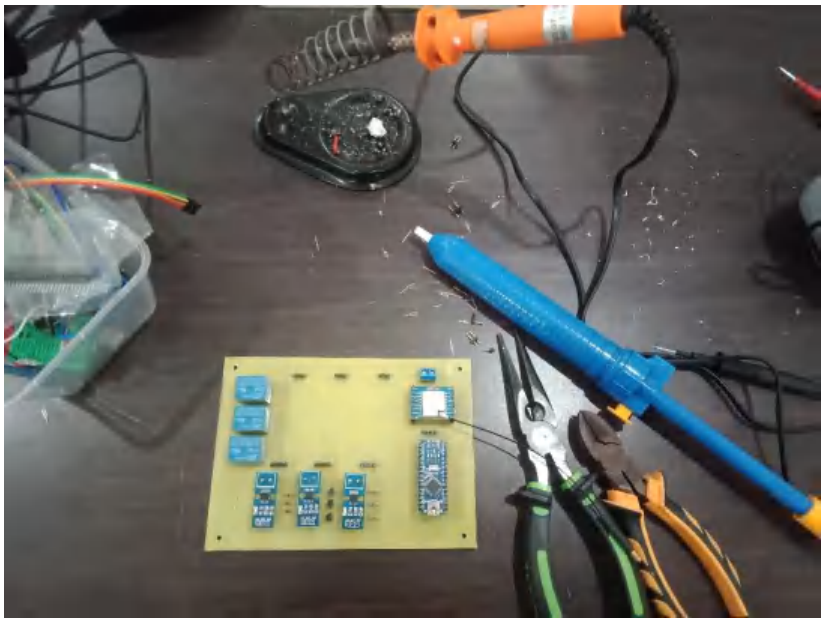
LAMPIRAN

Lampiran 1 Proses uji coba komponen sebelum membuat skematik dan PCB layout.

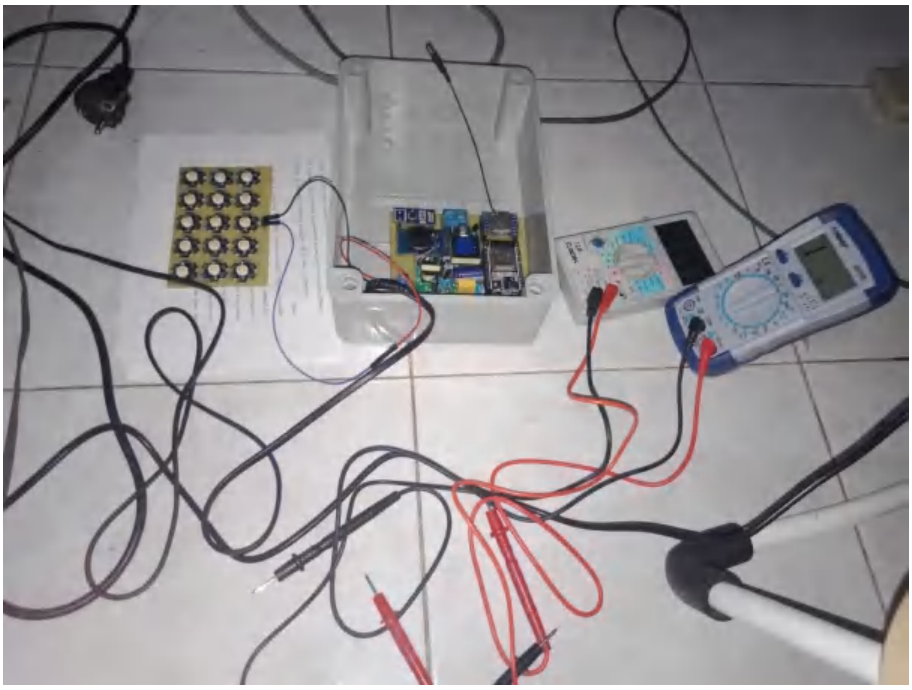


Lampiran 2 Proses fabrikasi layout PCB





Lampiran 3 Uji coba rangkaian



Lampiran 4 Pengujian dan pengambilan data

Lampiran 5 *Listing* program

```
String arusPath = "/ARUS";
String dayaPath = "/DAYA";
String pwmPath = "/PWM";
String ldrPath = "/LDR";
String mode = "a";
String pwmControl = "0|0|0";
String broadcastData = mode+pwmControl;
float ldr1, ldr2, ldr3 = 0;
String saveLogServer = "https://streetlightiot.000webhostapp.com/saveLog.php";
bool forceSaveLog = false;

struct DataArray {
  String val[6];
};

// Inisialisasi WiFi
void initWiFi() {
  Serial.println("Mencari WiFi...");
  WiFi.mode(WIFI_STA);
  int nbVisibleNetworks = WiFi.scanNetworks();
  while (nbVisibleNetworks == 0) {
    Serial.println("Tidak ada WiFi ditemukan, coba ulang setelah 1 detik...");
    delay(1000);
    nbVisibleNetworks = WiFi.scanNetworks();
  }
  boolean wifiFound = false;
  int i, n;
  for (i = 0; i < nbVisibleNetworks; ++i) {
    for (n = 0; n < KNOWN_SSID_COUNT; n++) {
      if (!strcmp(KNOWN_SSID[n], WiFi.SSID(i).c_str())) {
        wifiFound = true;
        break;
      }
    }
    if (wifiFound) break;
  }
  if (!wifiFound) {
```

```
    Serial.println("Tidak ada WiFi yang dikenali");
    while (1);
  }
  Serial.print("Menghubungkan ke " + String(KNOWN_SSID[n]) + "...");
  WiFi.begin(KNOWN_SSID[n], KNOWN_PASSWORD[n]);
  while (WiFi.status() != WL_CONNECTED) {
      delay(500);
      Serial.print(".");
  }
  Serial.println(" BERHASIL");
}

// Inisialisasi Firebase
void initFirebase() {
  Serial.print("Menghubungkan ke Firebase...");
  fconfig.api_key = API_KEY;
  fconfig.database_url = DATABASE_URL;
  fauth.user.email = USER_EMAIL;
  fauth.user.password = USER_PASSWORD;
  Firebase.begin(&fconfig, &fauth);
  Firebase.reconnectWiFi(true);
  if (Firebase.RTDB.setString(&fbdo, "config/mode", mode) &&
Firebase.RTDB.setString(&fbdo, "config/pwm", pwmControl)) {
    Serial.println(" BERHASIL");
  } else {
    Serial.println(" GAGAL");
    while(1);
  }
}

// Sinkronisasi Waktu
void syncTime() {
  struct tm timeinfo;
  int yr = 0;
  int mt = 0;
  int dy = 0;
  int hr = 0;
```

```
  int mi = 0;
  int se = 0;
  Serial.print("Menyinkronkan waktu...");
  configTime(gmtOffset_sec, daylightOffset_sec, ntpServer);
  getLocalTime(&timeinfo);
  Serial.println(&timeinfo, " %A, %B %d %Y %H:%M:%S");
  yr = timeinfo.tm_year + 1900;
  mt = timeinfo.tm_mon + 1;
  dy = timeinfo.tm_mday;
  hr = timeinfo.tm_hour;
  mi = timeinfo.tm_min;
  se = timeinfo.tm_sec;
  rtc.adjust(DateTime(yr, mt, dy, hr, mi, se));
}

// Simpan Data ke Firebase
void saveDataFirebase() {
  DataArray data1 = parseData(dataNode1);
  DataArray data2 = parseData(dataNode2);
  DataArray data3 = parseData(dataNode3);
  keyPath = "NODE_1";
  fjson.set(keyPath+teganganPath.c_str(), data1.val[0].toFloat());
  fjson.set(keyPath+arusPath.c_str(), data1.val[1].toFloat());
  fjson.set(keyPath+dayaPath.c_str(), data1.val[2].toFloat());
  fjson.set(keyPath+pwmPath.c_str(), data1.val[3].toFloat());
  fjson.set(keyPath+ldrPath.c_str(), data1.val[4].toFloat());
  ldr1 = data1.val[4].toFloat();
  keyPath = "NODE_2";
  fjson.set(keyPath+teganganPath.c_str(), data2.val[0].toFloat());
  fjson.set(keyPath+arusPath.c_str(), data2.val[1].toFloat());
  fjson.set(keyPath+dayaPath.c_str(), data2.val[2].toFloat());
  fjson.set(keyPath+pwmPath.c_str(), data2.val[3].toFloat());
  fjson.set(keyPath+ldrPath.c_str(), data2.val[4].toFloat());
  ldr2 = data2.val[4].toFloat();
  keyPath = "NODE_3";
  fjson.set(keyPath+teganganPath.c_str(), data3.val[0].toFloat());
  fjson.set(keyPath+arusPath.c_str(), data3.val[1].toFloat());
```

```
  fjson.set(keyPath+dayaPath.c_str(), data3.val[2].toFloat());
  fjson.set(keyPath+pwmPath.c_str(), data3.val[3].toFloat());
  fjson.set(keyPath+ldrPath.c_str(), data3.val[4].toFloat());
  ldr3 = data3.val[4].toFloat();
  Firebase.RTDB.setJSON(&fbdo, "/data/", &fjson);
  if (data1.val[5]=="save" || data2.val[5]=="save" || data3.val[5]=="save") {
    forceSaveLog = true;
  }
}

// Baca Data dari Firebase
void readDataFirebase() {
  if (Firebase.RTDB.getJSON(&fbdo, "/config")) {
    FirebaseJson json = fbdo.to<FirebaseJson>();
    FirebaseJsonData result;
    mode = (json.get(result, "mode")) ? result.to<String>() : mode;
    if (mode == "a") {
      if (now.hour() >= 18 && now.hour() <= 23) {       // Jam >= 18 DAN <= 23
(Kondisi yang memenuhi: 18.00 - 23.59)
        pwmControl = "1|1|1";
      } else if (now.hour() >= 0 && now.hour() <= 5) {  // Jam >= 0 DAN <=
5   (Kondisi yang memenuhi: 00.00 - 05.59)
        pwmControl = "0.5|0.5|0.5";
        if (now.hour() == 5 && now.minute() > 30) {     // Jam lewat 05.30, lampu
mati
          pwmControl = "0|0|0";
        }
      } else {                                          // Selain
itu          (Kondisi yang memenuhi: 06.00 - 17.59)
        pwmControl = "0|0|0";
      }
    } else {
      pwmControl = (json.get(result, "pwm")) ? result.to<String>() : pwmControl;
    }
    broadcastData = mode+pwmControl;
  }
}
```

```
void saveLog() {
  HTTPClient http;
  http.begin(saveLogServer.c_str());
  int httpResponseCode = http.GET();
  if (httpResponseCode == 200) {
    Serial.println(" BERHASIL");
  } else {
    Serial.println(" GAGAL");
  }
  http.end();
}

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

  initWiFi();
  initFirebase();

  if (!rtc.begin()) {
    Serial.println("Tidak ada RTC terpasang");
    while (1);
  }

  syncTime();

  LoRa.setPins(csPin, resetPin, irqPin);
  if (!LoRa.begin(frequency)) {
    Serial.println("Inisialisasi LoRa gagal");
    while (1);
  }
  LoRa.onReceive(onReceive);
  LoRa.onTxDone(onTxDone);
  LoRa_rxMode();

  Serial.println("READY");
```

```
}

void loop() {
  now = rtc.now();
  if (runEvery(10000)) {
    char formatRTC[] = "YYYYMMDD-hh:mm:ss";

    saveDataFirebase();
    readDataFirebase();

    Serial.println("=======================================");
    Serial.println(now.toString(formatRTC));

    Serial.println("Node #1: " + dataNode1);
    Serial.println("Node #2: " + dataNode2);
    Serial.println("Node #3: " + dataNode3);

    Serial.println("CONFIG: " + mode + " " + pwmControl);
    Serial.println("=======================================");
    Serial.println("");

    LoRa_sendMessage(broadcastData);
  }

  if (now.minute() != lastSaveLog.minute() || forceSaveLog) {
    forceSaveLog = false;
    lastSaveLog = now;
    Serial.println("--------------------");
    Serial.print("Saving LOG...");
    saveLog();
    Serial.println("--------------------");
    Serial.println("");
  }
}

void LoRa_rxMode(){
  LoRa.disableInvertIQ();
```

```
  LoRa.receive();
}

void LoRa_txMode(){
  LoRa.idle();
  LoRa.enableInvertIQ();
}

void LoRa_sendMessage(String message) {
  LoRa_txMode();
  LoRa.beginPacket();
  LoRa.print(message);
  LoRa.endPacket(true);
}

void onReceive(int packetSize) {
  String message = "";

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

  String idNode = message.substring(0, 1);
  String data = message.substring(1);
  if (idNode == "1") {
    dataNode1 = data;
  } else if (idNode == "2") {
    dataNode2 = data;
  } else if (idNode == "3") {
    dataNode3 = data;
  }
}

void onTxDone() {
  LoRa_rxMode();
}
```

```
boolean runEvery(unsigned long interval){
  static unsigned long previousMillis = 0;
  unsigned long currentMillis = millis();
  if (currentMillis - previousMillis >= interval)
  {
    previousMillis = currentMillis;
    return true;
  }
  return false;
}

DataArray parseData(String data) {
  DataArray dataArray;
  int indexArray = 0;
  while (data.length() > 0) {
    int separator = data.indexOf("|");
    if (separator == -1) {
      dataArray.val[indexArray++] = data;
      break;
    } else {
      dataArray.val[indexArray++] = data.substring(0, separator);
      data = data.substring(separator+1);
    }
  }
  return dataArray;
}
```

Kontrol lampu jalan

```
#include <SPI.h>
#include <LoRa.h>
#include <Wire.h>
#include <Adafruit_INA219.h>

Adafruit_INA219 inaSensor;

const int mosfetPin = 6;
int mosfetPWM = 0;
float mosfetPercent = mosfetPWM/255;

const int ldrPin = A1;
float ldrValue = 0;
float ldrThreshold = 0.3;

float voltage = 0;
float current = 0;
float power = 0;

const long frequency = 433E6;
const int csPin = 10;
const int resetPin = 9;
const int irqPin = 2;

String mode = "a";
String loraNode = "1";

bool forceSaveLog = false;
unsigned long currentTime = 0;
unsigned long previousOutputTime = 0;

void setup() {
  Serial.begin(115200);
  while (!Serial);
```

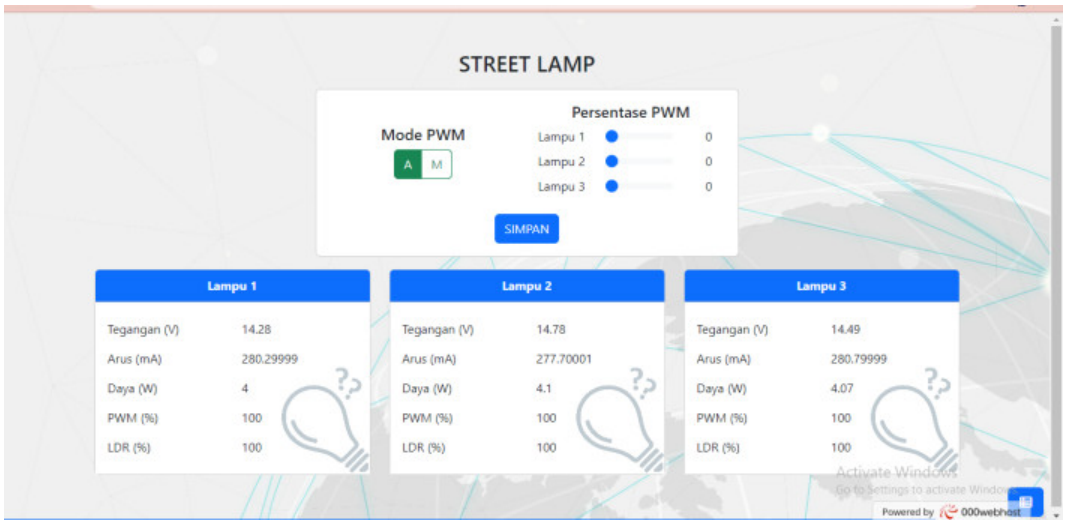
```
  pinMode(mosfetPin, OUTPUT);
  analogWrite(mosfetPin, mosfetPWM);

  if (!inaSensor.begin()) {
    Serial.println("Sensor INA219 tidak ditemukan");
    while(1);
  }

  LoRa.setPins(csPin, resetPin, irqPin);
  if (!LoRa.begin(frequency)) {
    Serial.println("Inisialisasi LoRa gagal");
    while (1);
  }
  LoRa.onReceive(onReceive);
  LoRa.onTxDone(onTxDone);
  LoRa_rxMode();

  Serial.println("READY");
}

void loop() {
  voltage = inaSensor.getBusVoltage_V();
  voltage = (voltage < 0) ? 0 : voltage;
  current = inaSensor.getCurrent_mA();
  current = (current < 0) ? 0 : current;
  power = voltage * (current/1000);
  power = (power < 0) ? 0 : power;
  mosfetPercent = (float)mosfetPWM / 255;
  ldrValue = (float)analogRead(ldrPin) / 1023;

  if (runEvery(3000)) {
    String message = loraNode;
    message += String(voltage, 2) + "|";
    message += String(current, 2) + "|";
    message += String(power, 2) + "|";
    message += String(mosfetPercent, 2) + "|";
    if (forceSaveLog) {
```

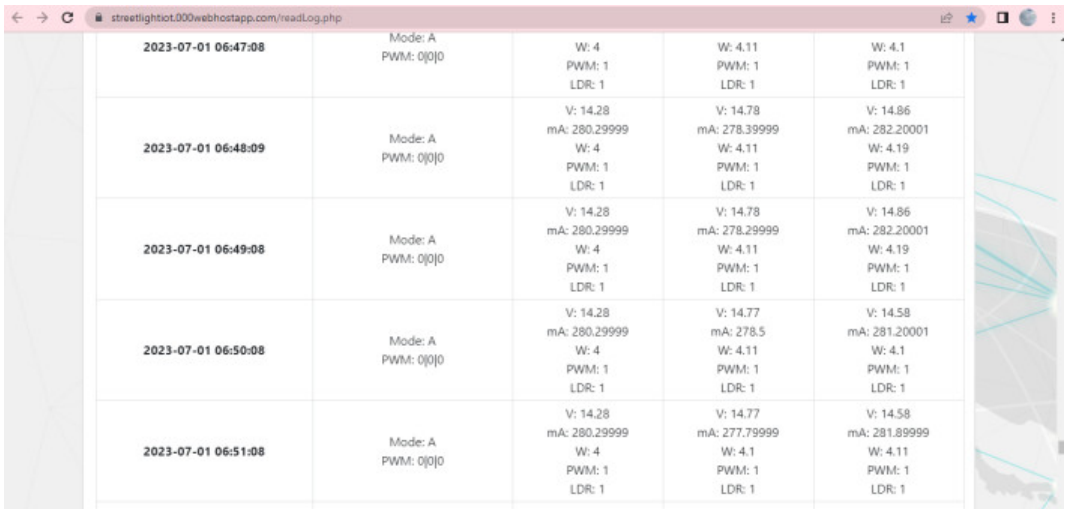
```
      message += String(ldrValue, 2) + "|";
      message += "save";
    } else {
      message += String(ldrValue, 2);
    }
    Serial.print("#"+loraNode+" ");
    Serial.println(message.substring(1));

    LoRa_sendMessage(message);
  }

  if (mode == "a" && ldrValue <= ldrThreshold) {
    mosfetPWM = ldrValue * 255;
    forceSaveLog = true;
    currentTime = millis();
    if (currentTime - previousOutputTime >= 1000) {
      previousOutputTime = millis();
      Serial.print("Mode LDR Aktif... ");
      Serial.println(String(voltage, 2) + "V|" + String(current, 2) + "ma|" +
String(power, 2) + "W|" + String(mosfetPercent*100, 2) + "%(PWM)|" +
String(ldrValue*100, 2) + "%(LDR)");
    }
  } else {
    forceSaveLog = false;
  }

  if (mosfetPWM > 255) {
    mosfetPWM = 255;
  } else if (mosfetPWM < 0) {
    mosfetPWM = 0;
  }

  analogWrite(mosfetPin, mosfetPWM);
}

void LoRa_rxMode(){
  LoRa.enableInvertIQ();
```

```
  LoRa.receive();
}

void LoRa_txMode(){
  LoRa.idle();
  LoRa.disableInvertIQ();
}

void LoRa_sendMessage(String message) {
  LoRa_txMode();
  LoRa.beginPacket();
  LoRa.print(message);
  LoRa.endPacket(true);
}

void onReceive(int packetSize) {
  String message = "";

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

  mode = message.substring(0, 1);
  String pwmControl = message.substring(1);
  Serial.println("CONFIG: " + mode + " " + pwmControl);
  int separator = pwmControl.indexOf("|");
  if (loraNode == "1") {
    pwmControl = pwmControl.substring(0, separator);
  } else if (loraNode == "2") {
    pwmControl = pwmControl.substring(separator+1);
    separator = pwmControl.indexOf("|");
    pwmControl = pwmControl.substring(0, separator);
  } else if (loraNode == "3") {
    pwmControl = pwmControl.substring(separator+1);
    separator = pwmControl.indexOf("|");
    pwmControl = pwmControl.substring(separator+1);
    separator = pwmControl.indexOf("|");
```

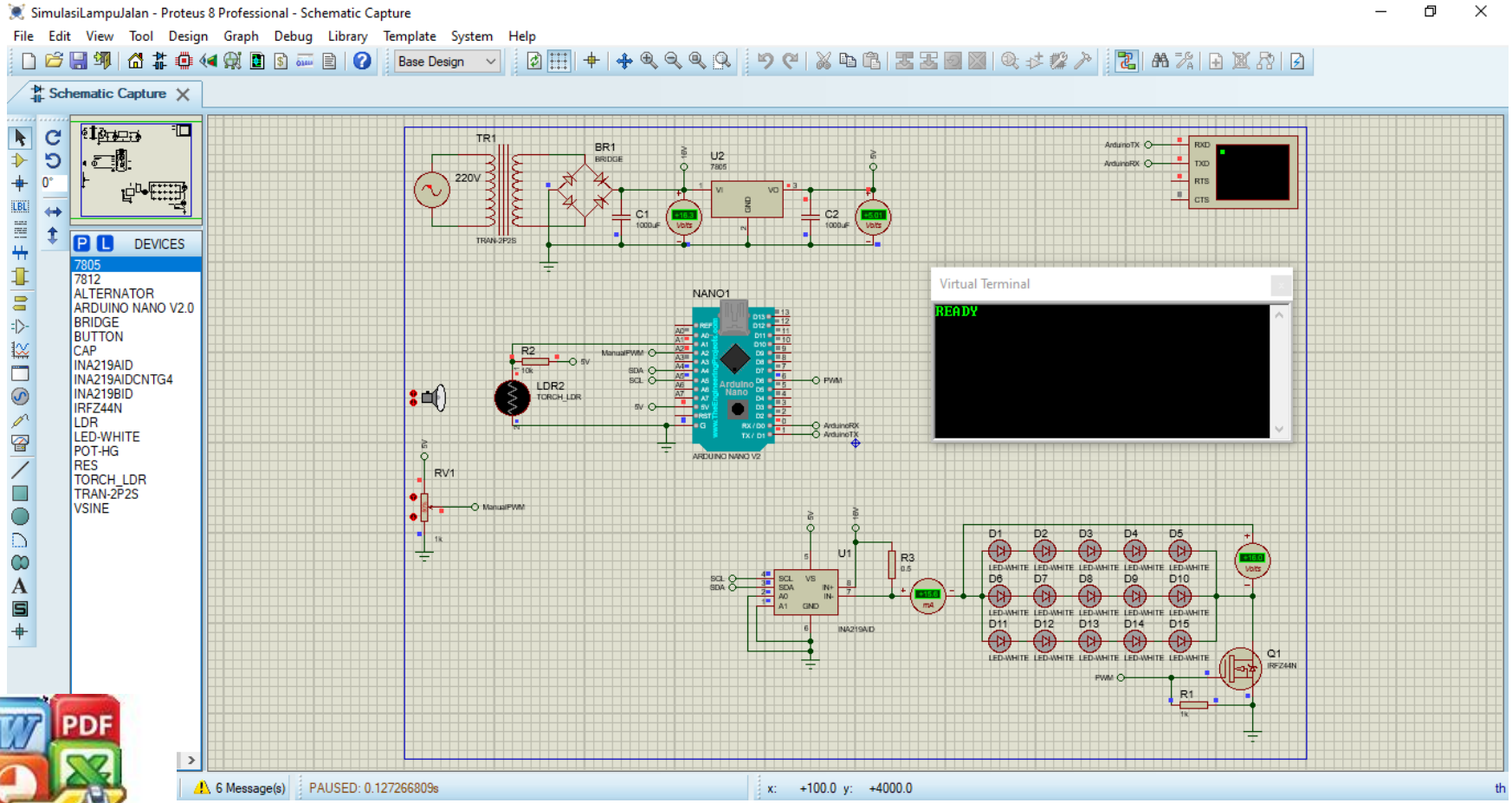
```
    pwmControl = pwmControl.substring(0, separator);
  }
  mosfetPWM = pwmControl.toFloat() * 255;
}

void onTxDone() {
  LoRa_rxMode();
}

boolean runEvery(unsigned long interval){
  static unsigned long previousMillis = 0;
  unsigned long currentMillis = millis();
  if (currentMillis - previousMillis >= interval)
  {
    previousMillis = currentMillis;
    return true;
  }
  return false;
}
```

Lampiran 6 *Monitoring* dan *database*

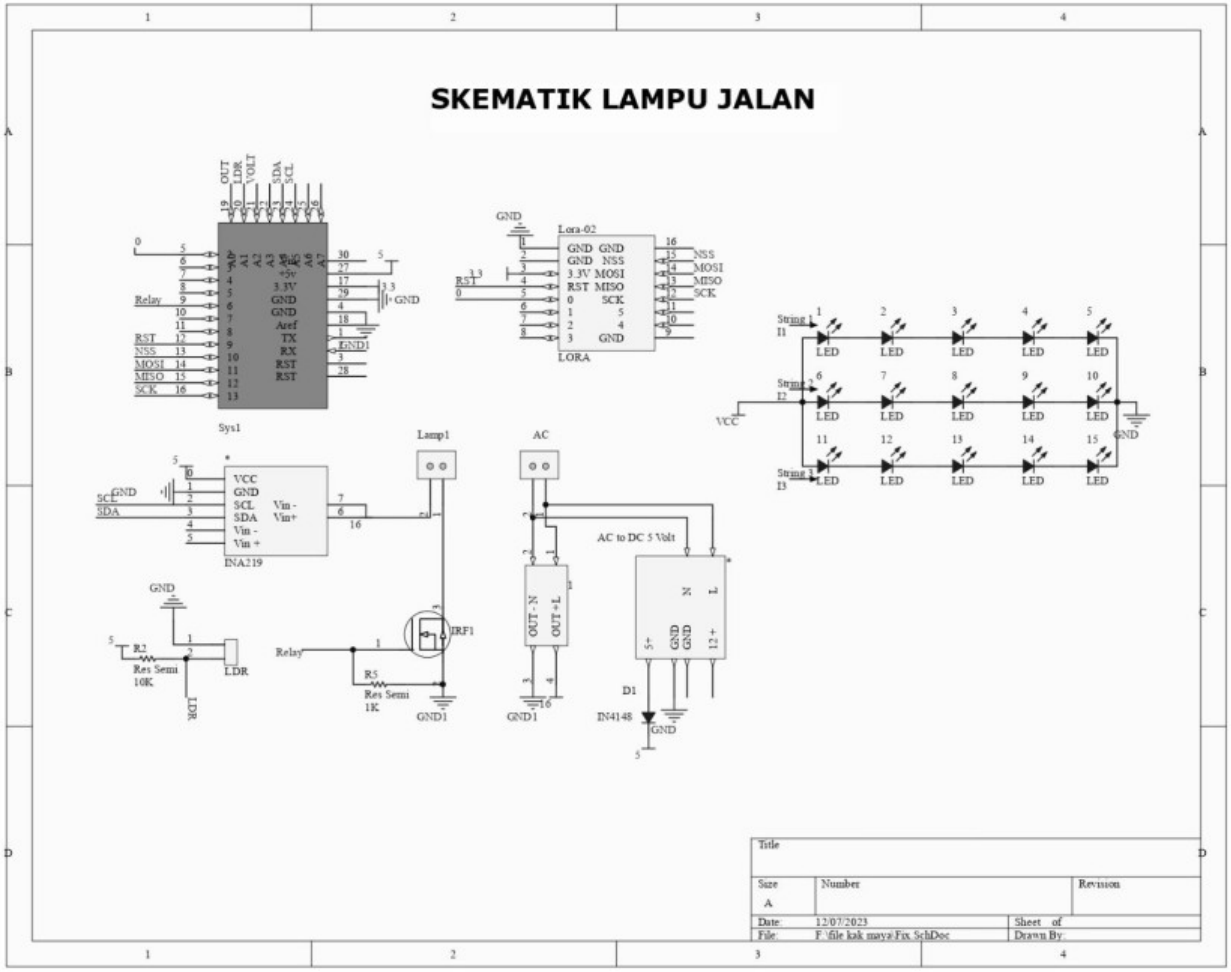




Lampiran 7 Rangkaian simulasi proteus 8 professional



Lampiran 8 Skematik lampu jalan



Lampiran 9 Skematik kontrol utama lampu jalan