

DAFTAR PUSTAKA

- Abuarqoub, A., Abusaimh, H., Hammoudeh, M., Uliyan, D., Abu-Hashem, M. A., Murad, S., ... & Al-Fayez, F. (2017). A survey on internet of things enabled smart campus applications. In *Proceedings of the International Conference on Future Networks and Distributed Systems* (pp. 1-7).
- Achmad, A., Hasanuddin, Z. B., Sadjad, R. S., Anshar, M., Muslimin, Z., Achmad, A. D., ... & Syarif, S. (2019). Implementasi Aplikasi Database Untuk Mendukung Sistem Smart card Di STKIP Muhammadiyah Bone. *JURNAL TEPAT: Teknologi Terapan untuk Pengabdian Masyarakat*, 2(1), 31-36.
- Al-Giffary, A. (2022). Desain Implementasi Modul Sensor Array Terkoneksi Server IoT Berbasis Antena Cerdas Untuk Mendukung Jaringan Smart Campus. Skripsi. Universitas Hasanuddin.
- Atzori, L., Iera, A., & Morabito, G. (2010). The internet of things: a survey. *Computer Networks*. doi, 10, 1016.
- Badan Pusat Statistik. (2020). Definisi Interoperabilitas [Interoperability]. <https://qasp2020.bps.go.id/posts/84f97df21abb4947afd5e271bb586f04/interoperability/definisi-interoperabilitas>
- Bormann, C., Castellani, A. P., & Shelby, Z. (2012). Coap: An application protocol for billions of tiny internet nodes. *IEEE Internet Computing*, 16(2), 62-67.
- Bröring, A., Echterhoff, J., Jirka, S., Simonis, I., Everding, T., Stasch, C., ... Lemmens, R. (2011). New generation Sensor Web Enablement. *Sensors*. 11(3), 2652-2699. <https://doi.org/10.3390/s110302652>
- Effendi, S., Heriansyah, H., Haryansyah, H., & Pamungkas, M. S. (2017). Rekayasa Sistem Parkir Berlangganan Berbasis RFID (Radio Frequency Identification). *Journal of Applied Microcontroller and Autonomous System*, 3(1), 27-37.
- Flores-Martin, D., Pérez-Vereda, A., Berrocal, J., Canal, C., & Murillo, J. M. (2018, December). Interconnecting IoT devices to improve the QoL of elderly people.

In *International Workshop on Gerontechnology* (pp. 83-93). Cham: Springer International Publishing.

Fortino, G., Savaglio, C., Palau, C. E., de Puga, J. S., Ganzha, M., Paprzycki, M., ... & Llop, M. (2018). Towards multi-layer interoperability of heterogeneous IoT platforms: The INTER-IoT approach. *Integration, interconnection, and interoperability of IoT systems*, 199-232.

Geetanjali, V., Subramanian, I., Kannan, G., Prathiba, S. B., & Raja, G. (2019). IoTexpert: Interconnection, interoperability, and integration of IoT platforms. In *Proceedings of the 11th International Conference on Advanced Computing, ICoAC 2019* (pp. 212–219). Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/ICoAC48765.2019.246842>

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7), 1645-1660.

Indonesia. (1999). *Undang-Undang Nomor 36 Tahun 1999 tentang Telekomunikasi*.

ITU. (2005). ITU Internet report 2005: The internet of things. *ITU Internet Report 2005*, 212.

Khatimah, K. (2022). Sistem Presensi Berbasis Smart Hybrid Reader. Skripsi. Universitas Hasanuddin.

Kinkar, S., Hennessy, M., & Ray, S. (2016). An ontology and integration framework for smart communities. *Journal of Computing and Information Science in Engineering*, 16(1), 011003.

Lueth, K. L. (2020). State of the IoT 2020: 12 billion IoT connections, surpassing non-IoT for the first time. <https://iot-analytics.com/state-of-the-iot-2020-12-billion-iot-connections-surpassing-non-iot-for-the-first-time/>

Madakam, S., Lake, V., Lake, V., & Lake, V. (2015). Internet of Things (IoT): A literature review. *Journal of Computer and Communications*, 3(05), 164.

- Muhamad, W., Kurniawan, N. B., & Yazid, S. (2017). Smart campus features, technologies, and applications: A systematic literature review. In *2017 International conference on information technology systems and innovation (ICITSI)* (pp. 384-391). IEEE.
- Mustaqbal, M. S., Firdaus, R. F., & Rahmadi, H. (2015). Pengujian aplikasi menggunakan black box testing boundary value analysis (studi kasus: Aplikasi prediksi kelulusan smnptn). *Jurnal Ilmiah Teknologi Infomasi Terapan*, 1(3).
- Navani, D., Jain, S., & Nehra, M. S. (2017, December). The internet of things (IoT): A study of architectural elements. In *2017 13th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)* (pp. 473-478). IEEE.
- Nur, A., Ismail, A., FS, A., Nuryana, A., & Informatika, T. (2019). Perancangan Website Data Karyawan Dengan menggunakan PHP dan MYSQL. *J. Sist. basis data*, no. January, 1-8.
- Prihatmoko, C.R. (2021). Pengembangan Teknologi Smart Hybrid Reader Untuk Sistem Smart Campus Unhas. Skripsi. Universitas Hasanuddin.
- Rahman, H., & Hussain, M. I. (2020). A comprehensive survey on semantic interoperability for Internet of Things: State-of-the-art and research challenges. *Transactions on Emerging Telecommunications Technologies*, 31(12), e3902.
- Ray, P. P. (2018). A survey on Internet of Things architectures. *Journal of King Saud University-Computer and Information Sciences*, 30(3), 291-319.
- Rozana, L., & Musfekar, R. (2020). Analisis Dan Perancangan Sistem Informasi Pengarsipan Surat Berbasis Web Pada Kantor Lurah Desa Dayah Tuha. *Cyberspace: Jurnal Pendidikan Teknologi Informasi*, 4(1), 14-20.
- Sabran, Purnamawati, & Nasruddin. (2020). Penerapan Smart Lab Automatitation Berbasis IoT Pada Laboratorium Digital Jurusan Pendidikan Teknik Elektronika FT-UNM. *JETC*, 15(2), 1–23.

- Sari, A. O., & Abdilah, A. Sunarti. (2015). *Web Programming*. Graha Ilmu. Yogyakarta.
- Secure Technology Alliance (2005). *Contactless Smart Card Technology and Electronic Passports: Frequently Asked Questions*. https://www.securetechalliance.org/resources/pdf/RFID_and_Contactless_Smart_Cards_FAQ_FINAL_042105.pdf
- Shobrina, U. J., Primananda, R., & Maulana, R. (2018). Analisis Kinerja Pengiriman Data Modul Transceiver NRF24I01, Xbee dan Wifi ESP8266 Pada Wireless Sensor Network. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(4), 1510-1517.
- Silva, J. D. C., Rodrigues, J. J. P. C., Saleem, K., Kozlov, S. A., & Rabelo, R. A. L. (2019). M4DN.IoT-A Networks and Devices Management Platform for Internet of Things. *IEEE Access*, 7, 53305–53313. <https://doi.org/10.1109/ACCESS.2019.2909436>
- Solichin, A., & Kom, S. (2005). Pemrograman WEB dengan PHP dan MySQL. *Achmatim.Net*. https://www.researchgate.net/publication/236885805_Pemrograman_Web_dengan_PHP_dan_MySQL.
- Srivastava, L. (2006, March). Pervasive, ambient, ubiquitous: the magic of radio. In *European Commission Conference ‘From RFID to the Internet of Things’*, Bruxelles, Belgium.
- Sterling, B. (2005). *Shaping things—Mediawork pamphlets*. https://mitp-content-server.mit.edu/books/content/sectbyfn/books_pres_0/5950/mediawork.zip/titles/shaping/shaping_book.html
- Sudarmanto, S., & Cahyani, A. (2007). Perancangan Sistem Pengendalian Motor Servo pada Robot Berkaki Menggunakan Microcontroller PIC 16F84. In *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*.
- Sutaya, I Wayan. (2014). *Sistem Mikroprosesor*. Graha Ilmu. Yogyakarta.
- Swetina, J., Lu, G., Jacobs, P., Ennesser, F., & Song, J. (2014). Toward a standardized common M2M service layer platform: Introduction to

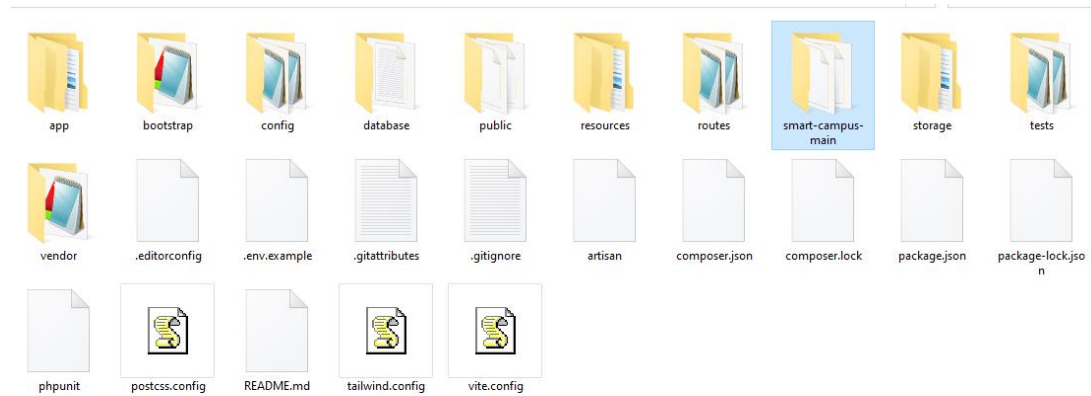
- oneM2M. *IEEE Wireless Communications*, 21(3), 20–26.
<https://doi.org/10.1109/MWC.2014.6845045>
- Szilagyi, I., & Wira, P. (2016). Ontologies and semantic Web for the internet of things - A survey. In *IECON Proceedings (Industrial Electronics Conference)* (pp. 6949–6954). IEEE Computer Society.
<https://doi.org/10.1109/IECON.2016.7793744>
- Tian, Z., Cui, Y., An, L., Su, S., Yin, X., Yin, L., & Cui, X. (2018). A real-time correlation of host-level events in cyber range service for smart campus. *IEEE Access*, 6, 35355–35364. <https://doi.org/10.1109/ACCESS.2018.2846590>
- Tjahjana, A., & Irawan, D. (2010, June). Sistem Smart Class Room Berbasis Smart Card Dan Bahasa Pemrograman C++. In *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*.
- Turang, D. A. O. (2015, December). Pengembangan sistem relay pengendalian dan penghematan pemakaian lampu berbasis mobile. In *Seminar Nasional Informatika (SEMNASIF)* (Vol. 1, No. 1).
- Vargas, D. C. Y., & Salvador, C. E. P. (2016). Smart IoT gateway for heterogeneous devices interoperability. *IEEE Latin America Transactions*, 14(8), 3900–3906. <https://doi.org/10.1109/TLA.2016.7786378>.
- Xu, Z., Chao, L., & Peng, X. (2018). T-REST: An open-enabled architectural style for the Internet of Things. *IEEE Internet of Things Journal*, 6(3), 4019–4034.
- Yan, Z., Zhang, P., & Vasilakos, A. V. (2014). A survey on trust management for Internet of Things. *Journal of Network and Computer Applications*, 42, 120–134. <https://doi.org/10.1016/j.jnca.2014.01.014>
- Yang, A. M., Li, S. S., Ren, C. H., Liu, H. X., Han, Y., & Liu, L. (2018). Situational awareness system in the smart campus. *Ieee Access*, 6, 63976–63986. doi:10.1109/ACCESS.2018.2877428.
- Zhamanov, A., Sakhiyeva, Z., Suliyev, R., & Kaldykulova, Z. (2018). IoT smart campus review and implementation of IoT applications into education process

of university. In *2017 13th International Conference on Electronics, Computer and Computation, ICECCO 2017* (Vol. 2018-January, pp. 1–4). Institute of Electrical and Electronics Engineers Inc.
<https://doi.org/10.1109/ICECCO.2017.8333334>

LAMPIRAN

Lampiran 1. Program Code Website

Kode program dapat diakses pada link <https://bit.ly/CodeWebsiteTA-Miya>



Lampiran 2. Program Arduino untuk ESP8266

Kode Program mikrokontroler untuk sistem smart campus dapat diakses pada link https://bit.ly/codemicrocontroller_TAMiya

Attendance System

```
#include <SPI.h>
#include <MFRC522.h>
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
#include <ArduinoJson.h> // Untuk mengurai JSON
#include <WiFiManager.h>
#define LED_STDBY 4 // led standby
#define LED_FALSE 5 // led False
#define LED_TRUE 15 // led true
#define SDA_PIN 2 //D4
#define RST_PIN 0 //D3

const char* ssid = "Realmee";
const char* password = "12345679";
```



```

const char* serverUrl = "http://smartcampusunhas.com/api/handle-attendance"; //
Replace with your server URL

const char* WebURL = "http://smartcampusunhas.com/api/attendaces";

const char* mac_system; // dari giga 98-29-A6-93-3C-BB

String key = "{\"message\":\":"
            "0"
            "\"}";

String idMac;

int id_new = 0;

short perintah = 3;

unsigned long previousMillis = 0; // Waktu sebelumnya
unsigned long interval = 1000; // Interval waktu (1 detik)
MFRC522 mfrc522(SDA_PIN, RST_PIN);

void setup() {
  Serial.begin(115200);
  Serial.println();
  Serial.println(WiFi.macAddress());
  idMac = WiFi.macAddress();

  WiFiManager wifiManager;
  wifiManager.autoConnect("Smart Campus-Absensi"); // "ESP8266-AP" adalah
nama AP WiFi yang akan dibuat jika gagal terhubung ke WiFi sebelumnya
  Serial.println("Terhubung ke jaringan WiFi!");
  Serial.print("SSID: ");
  Serial.println(WiFi.SSID());

  pinMode(LED_STDBY, OUTPUT);
  pinMode(LED_FALSE, OUTPUT);
  pinMode(LED_TRUE, OUTPUT);
  SPI.begin();

```

```

mfr522.PCD_Init();
Serial.println("Dekatkan Kartu RFID Anda ke Reader");
Serial.println();
digitalWrite(LED_STDBY, HIGH);
digitalWrite(LED_FALSE, LOW);
digitalWrite(LED_TRUE, LOW);
}

void loop() {
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
  unsigned long currentMillis = millis();
  unsigned long currentMicros = micros(); // Waktu mikrodetik saat ini
  if (currentMillis - previousMillis >= interval) {
    web();
    previousMillis = currentMillis;
  }

  // if (currentMicros - previousMicros >= intervalMicros) {
  if (mfr522.PICC_IsNewCardPresent() && mfr522.PICC_ReadCardSerial()) {
    kirimid();
  }
  // previousMicros = currentMicros;
  // }

  switch (perintah) {
  case 1:
    digitalWrite(LED_STDBY, LOW);

```

```
digitalWrite(LED_FALSE, HIGH);
digitalWrite(LED_TRUE, LOW);
Serial.println("1");
delay(500);
digitalWrite(LED_STDBY, HIGH);
digitalWrite(LED_FALSE, LOW);
digitalWrite(LED_TRUE, LOW);
perintah = 0;
break;
case 2:
digitalWrite(LED_STDBY, LOW);
digitalWrite(LED_FALSE, LOW);
digitalWrite(LED_TRUE, HIGH);
Serial.println("2");
delay(5000);
digitalWrite(LED_STDBY, HIGH);
digitalWrite(LED_FALSE, LOW);
digitalWrite(LED_TRUE, LOW);
perintah = 0;
break;
}
}

void web() {
WiFiClient client;
HTTPClient http;
http.begin(client, WebURL);

// Mengeksekusi permintaan dan menyimpan hasil
int httpCode = http.GET();
if (httpCode == 200) {
```

```
if (httpCode == HTTP_CODE_OK) {
    String payload = http.getString();
    Serial.println("Data JSON diterima:");
    Serial.println(payload);

    // Mengurai data JSON
    DynamicJsonDocument doc(1024); // Ubah sesuai kebutuhan
    deserializeJson(doc, payload);

    // Memeriksa setiap objek dalam JSON
    JsonArray data = doc.as<JsonArray>();
    for (JsonObject item : data) {
        int id = item["id"];
        String mac = item["mac"];
        String status = item["status"];
        // Menampilkan data per ID
        // Serial.println("ID: " + String(id));
        // Serial.println("MAC: " + mac);
        // Serial.println("Status: " + status);
        // Serial.println();

        // Melakukan sesuatu dengan data yang diambil
        if (id > id_new) {
            if (mac == idMac) {
                if (status == "0") {
                    Serial.println("terbuka");
                    perintah = 2;
                }
                if (status == "1") {
                    Serial.println("id salah");
                    perintah = 1;
                }
            }
        }
    }
}
```

```

        }
    }
    id_new = id;
}
}
}
} else {
    Serial.println("Gagal mengambil data dari endpoint.");
}
Serial.println(id_new);

http.end();
client.stop();
}

void kirimid() {
    WiFiClient client;
    HTTPClient http;
    http.begin(client, serverUrl);
    Serial.print("UID tag :");
    String rfid = "";
    for (byte i = 0; i < mfrc522.uid.size; i++) {
        rfid += mfrc522.uid.uidByte[i];
    }
    Serial.println(rfid);
    http.addHeader("Content-Type", "application/json");
    String labarea_id = "1";
    String postData = "{\"rfid\":\"" + rfid + "\",\"labarea_id\":\"" + labarea_id + "\"}";
    int httpResponseCode = http.POST(postData);
    delay(1000);
    String response = http.getString();
}

```

```

Serial.println("Response: " + response);
// Serial.println("HTTP Response code: " + String(httpResponseCode));
if (response == key) {
    perintah = 2;
} else if (response == "1") {
    perintah = 1;
}
http.end();
client.stop();
}

```

Parking System (Verification ID)

```

#include <SPI.h>
#include <MFRC522.h>
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
#include <ArduinoJson.h>
#include <WiFiManager.h>

#define LED_STDBY 4 // led standby
#define LED_FALSE 5 // led False
#define LED_TRUE 15 // led true
#define SDA_PIN 2 //D4
#define RST_PIN 0 //D3

const char* serverUrl = "http://smartcampusunhas.com/api/handle-action"; //
Replace with your server URL

const char* WebURL = "http://smartcampusunhas.com/api/commands";

String serverIP; //192.168.43.177

const int serverPort = 80;

String mac_system; // dari giga 98-29-A6-93-3C-BB

// String respon = ("message": "open");

```

```

// char perintah;
int id_new = 0;
short perintah = 3;

unsigned long previousMillis = 0; // Waktu sebelumnya
unsigned long interval = 3000; // Interval waktu (1 detik)
MFRC522 mfrc522(SDA_PIN, RST_PIN);

void setup() {
  Serial.begin(115200);
  Serial.println();
  Serial.println(WiFi.macAddress());
  mac_system = WiFi.macAddress();
  WiFiManager wifiManager;
  wifiManager.autoConnect("Smart Campus-Barrier Send"); // "ESP8266-AP"
  adalah nama AP WiFi yang akan dibuat jika gagal terhubung ke WiFi sebelumnya
  Serial.println("Terhubung ke jaringan WiFi!");
  Serial.print("SSID: ");
  Serial.println(WiFi.SSID());
  serverIP = WiFi.localIP();
  Serial.println(serverIP); // mencetak ip perangkat yang terdaftar
  //defenisikan PIN
  pinMode(LED_STDBY, OUTPUT);
  pinMode(LED_FALSE, OUTPUT);
  pinMode(LED_TRUE, OUTPUT);

  //perintah RFID
  SPI.begin();
  mfrc522.PCD_Init();
  Serial.println("Dekatkan Kartu RFID Anda ke Reader");
  Serial.println();

```

```
digitalWrite(LED_STDBY, HIGH);
digitalWrite(LED_FALSE, LOW);
digitalWrite(LED_TRUE, LOW);
}

void loop() {
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
  unsigned long currentMillis = millis();
  if (currentMillis - previousMillis >= interval) {
    web();
    previousMillis = currentMillis;
  }
  // kirim data dan koneksi RFID/ handle-action
  if (mfrc522.PICC_IsNewCardPresent() && mfrc522.PICC_ReadCardSerial()) {
    kirimid();
  }

  switch (perintah) {
    case 1:
      digitalWrite(LED_STDBY, LOW);
      digitalWrite(LED_FALSE, HIGH);
      digitalWrite(LED_TRUE, LOW);
      Serial.println("1");
      delay(500);
      digitalWrite(LED_STDBY, HIGH);
      digitalWrite(LED_FALSE, LOW);
      digitalWrite(LED_TRUE, LOW);
```



```

    perintah = 0;
    break;
case 2:
    digitalWrite(LED_STDBY, LOW);
    digitalWrite(LED_FALSE, LOW);
    digitalWrite(LED_TRUE, HIGH);
    Serial.println("2");
    sendDataToReceiver();
    delay(5000);
    digitalWrite(LED_STDBY, HIGH);
    digitalWrite(LED_FALSE, LOW);
    digitalWrite(LED_TRUE, LOW);
    perintah = 0;
    break;
}
}

void web() {
    WiFiClient client;
    HTTPClient http;
    http.begin(client, WebURL); //WebURL merupakan endpoint
    int httpCode = http.GET(); // mengirim request/get
    if (httpCode == 200) {
        String payload = http.getString(); // mengambil data dari endpoint
        DynamicJsonDocument doc(1024); // Ubah ukuran sesuai
        kebutuhan
        DeserializationError error = deserializeJson(doc, payload); // mengurai data dari
        endpoint
        if (error) {
            Serial.print("JSON parsing error: ");
            Serial.println(error.c_str());

```

```

} else {
    JSONArray data = doc.as<JSONArray>(); // data yang diurai
    for (JsonObject item : data) {
        int id = item["id"];
        String mac = item["mac"];
        int status = item["status"];
        if (id > id_new) { // memproses data yang telah di urai
            if (mac == mac_system) {
                if (status == 1) {
                    Serial.println("terbuka"); // jika data sesuai dengan perangkat
                    perintah = 2;
                    id_new = id;
                }
                if (status == 0) {
                    Serial.println("id salah"); // jika data tidak sesuai dengan perangkat
                    perintah = 1;
                }
            }
        }
    }
}

http.end(); // mengakhiri koneksi http
}

void kirimid() {
    WiFiClient client;
    HTTPClient http;
    http.begin(client, serverUrl); //WebURL merupakan endpoint untuk request
    Serial.print("UID tag :");
    String rfid = "";
    for (byte i = 0; i < mfrc522.uid.size; i++) { //memeriksa RFID yang terdeteksi

```

```

    rfid += mfr522.uid.uidByte[i];
}
Serial.println(rfid);
// Set headers (optional)
http.addHeader("Content-Type", "application/json");
;
String code_info = "1";
String areaparkir_id = "1";
// Your JSON data to send in the POST request
String postData = "{\"mac\": \"" + mac_system + "\", \"rfid\": \"" + rfid +
"\", \"code_info\": \"" + code_info + "\", \"areaparkir_id\": \"" + areaparkir_id + "\"}";
//menyatukan data yang diPOST
// Send HTTP POST request
int httpResponseCode = http.POST(postData); // POST data ke endpoint request
String response = http.getString(); // mengambil respon dari POST request
Serial.println("Response: " + response);
Serial.println("HTTP Response code: " + String(httpResponseCode));
if (httpResponseCode == 200) { // Mengolah respon dari POST request
    perintah = 2; // jika RFID terdaftar/benar
} else if (httpResponseCode == 404) { // jika RFID salah/tidak terdaftar
    perintah = 1;
} else {
    Serial.println("HTTP POST request failed");
}
http.end();
client.stop();
}

```

```

void sendDataToReceiver() {
    WiFiClient client;
    if (client.connect(serverIP, serverPort)) { //menghubungkan ke socket server

```

```

Serial.println("Terhubung ke Server");
client.println("2"); // mengirim data ke ESP penerima
client.println();
delay(1000);
while (client.available()) {
  String line = client.readStringUntil('\r');
  Serial.println(line);
}
client.stop();
Serial.println("Koneksi ditutup");
} else {
  Serial.println("Gagal terhubung ke Server");
}
}

```

Parking System (Barrier Gate)

```

#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <WiFiManager.h>

const int serverPort = 80;
//barier
int angle = 90;
const int stepsPerRevolution = 200; // Jumlah langkah pada satu putaran
const int stepPin = 12;           // Pin untuk sinyal step
const int dirPin = 14;           // Pin untuk sinyal arah (direction)
const int rotPin = 5;            // Pin untuk sinyal arah (direction)

// Buat objek server
WiFiServer server(serverPort);

```

```

void setup() {
  Serial.begin(115200);
  WiFiManager wifiManager;
  wifiManager.autoConnect("Smart Campus-Barrier Load"); // membuat ssid
  perangkat untuk menghubungkan ke wifi
  Serial.println("Terhubung ke jaringan WiFi!");
  Serial.print("SSID: ");
  Serial.println(WiFi.SSID());
  Serial.println(WiFi.localIP()); // mencetak ip perangkat yang terdaftar
  pinMode(stepPin, OUTPUT);
  pinMode(dirPin, OUTPUT);
  pinMode(rotPin, INPUT);
  server.begin(); // membuat server penerima data dari esp
  Serial.println("Server berjalan di port 80");
}

```

```

void loop() {
  short perintah = 3;
  WiFiClient client = server.available();
  if (client) {
    Serial.println("Client terhubung");
    String request = client.readStringUntil('\r'); //membaca data yang diterima
    Serial.println("Data yang diterima: " + request);
    request = perintah;
    if (perintah = 2) { //mengolah data yang diterima
      rotateClockwise(angle);
      perintah = 3;
    }
  }
}

void rotateClockwise(int angle) {

```

```

int steps = stepsPerRevolution * angle / 360; // Menghitung jumlah langkah
digitalWrite(dirPin, HIGH);
Serial.println(steps); // Set arah putaran menjadi searah jarum jam

for (int i = 0; i < steps; i++) {
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(10000);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(10000);
    // Serial.println("ok");
}

// Mengatur sudut kembali ke 0 setelah 1 detik
delay(1000);
digitalWrite(dirPin, LOW); // Set arah putaran menjadi berlawanan arah jarum
jam

for (int i = 0; i < steps; i++) {
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(10000);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(10000);
}
}

```

Indoor Monitoring

```

#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <WiFiManager.h>
#include <ESP8266HTTPClient.h>
#include <Adafruit_GFX.h>

```

```

#include <Adafruit_ST7789.h>
#include <DHT.h> /

#define TFT_DC D1 // TFT DC pin is connected to NodeMCU pin D1
(GPIO5)
#define TFT_RST D2 // TFT RST pin is connected to NodeMCU pin D2
(GPIO4)
#define TFT_CS D8
#define DHTPIN 12
#define DHTTYPE DHT11
Adafruit_ST7789 tft = Adafruit_ST7789(TFT_CS, TFT_DC, TFT_RST);
DHT dht11(DHTPIN, DHTTYPE);
const char* ssid = "Realmee";
const char* password = "12345679";
const char* serverUrl = "http://smartcampusunhas.com/api/indoormonitoring";
unsigned long previousMillis = 0;
unsigned long interval = 3000;

float suhu;
float lembab;
void setup(void) {
  Serial.begin(115200);
  WiFiManager wifiManager;
  wifiManager.autoConnect("Smart Campus-Indoor");
  Serial.println("Terhubung ke jaringan WiFi!");
  Serial.print("SSID: ");
  Serial.println(WiFi.SSID());
  Serial.print("IP Address: ");
  Serial.println(WiFi.localIP());

  tft.init(240, 240, SPI_MODE2);

```

```
tft.setRotation(2);

tft.fillScreen(ST77XX_BLACK);

tft.setTextWrap(false);
tft.setTextColor(ST77XX_BLUE, ST77XX_BLACK);
tft.setTextSize(3);
tft.setCursor(15, 40);
tft.print("TEMPERATURE:");
tft.setTextColor(ST77XX_BLUE, ST77XX_BLACK);
tft.setCursor(43, 140);
tft.print("HUMIDITY:");
tft.setTextSize(4);
dht11.begin();
}

char _buffer[7];
void loop() {
  tft.invertDisplay(true);
  delay(1000);

  int Humi = dht11.readHumidity() * 10;

  int Temp = dht11.readTemperature() * 10;

  tft.setTextColor(ST77XX_CYAN, ST77XX_BLACK); d
  if (Temp < 0)
    sprintf(_buffer, "-%02u.%1u", (abs(Temp) / 10) % 100, abs(Temp) % 10);
  else
    sprintf(_buffer, "%02u.%1u", (Temp / 10) % 100, Temp % 10);
```



```

tft.setCursor(26, 71);
tft.print(_buffer);
tft.drawCircle(161, 77, 4, ST77XX_CYAN);
tft.drawCircle(161, 77, 5, ST77XX_CYAN);
tft.setCursor(170, 71);
tft.print("C");

tft.setTextColor(ST77XX_CYAN, ST77XX_BLACK);
sprintf(_buffer, "%02u.%1u %%", (Humi / 10) % 100, Humi % 10);
tft.setCursor(50, 171);
tft.print(_buffer);
suhu = (Temp / 10.0);
lembab = (Humi / 10.0);
Serial.print(suhu, 1);
Serial.println(lembab, 1);

unsigned long currentMillis = millis();
if (currentMillis - previousMillis >= interval) {
    kirim();
    previousMillis = currentMillis;
}
}

void kirim() {
    WiFiClient client;
    HTTPClient http;
    http.begin(client, serverUrl);
    http.addHeader("Content-Type", "application/json");
    String postData = "{\"suhu\": \"" + String(suhu, 1) + "\", \"kelembaban\": \"" +
String(lembab, 1) + "\"}";
    int httpResponseCode = http.POST(postData);

```

```
// delay(1000);  
String response = http.getString();  
Serial.println("Response: " + response);  
  
http.end();  
}  
  
// end of code.
```