DAFTAR PUSTAKA

Abdul Hakim Prima Yuniarto, Lestiyanti, Y., Asrori, M. F., Laela, N., & Nurcholis, A. (2023). Perancangan Smart Door Lock System dengan Multi Sensor untuk Sistem Keamanan Rumah. *Techné : Jurnal Ilmiah Elektroteknika*, *22*(2), 333–342. https://doi.org/10.31358/techne.v22i2.404

Alfian Saputra, R., Dito Ridwansyah, R., Alfiana Erlangga, D., & Rilvani, E. (2025). Keamanan Sistem Operasi Dalam Era Internet of Things. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *9*(2), 1939–1944. https://doi.org/10.36040/jati.v9i2.12837

Amin, F., Abbasi, R., Mateen, A., Ali Abid, M., & Khan, S. (2022). A Step toward Next-Generation Advancements in the Internet of Things Technologies. *Sensors*, *22*(20). https://doi.org/10.3390/s22208072

Ardiansah, A., Nuraeni, M., Ridwang, & Adriani. (2024). Rancang Bangun Akses Kunci Pintu Otomatis menggunakan Fingerprint Berbasis Internet of Things (IoT). *VERTEX ELEKTRO Jurnal Teknik Elektro UNISMUH*, *16*(2), 31–39.

Arifin, W., Fitriansyah, A., & Setiadi, D. (2022). Pembatasan Akses Secara Fisik Dengan Sistem Fingerprint Doorlock Menggunakan Microcontroller Arduino Uno R3. *Jeis: Jurnal Elektro Dan Informatika Swadharma*, *2*(2), 81–88. https://doi.org/10.56486/jeis.vol2no2.234

Aryatama, F. A., & Samsugi, S. (2024). Sistem Keamanan Kendaraan Bermotor Dengan ESP32 Menggunakan Kontrol Android. *Smatika Jurnal*, *14*(01), 167–181. https://doi.org/10.32664/smatika.v14i01.1267

Azis, A., Nurdiana, N., & Saputra, J. (2024). Perancangan Prototipe Pengendali Pintu Pagar Otomatis Menggunakan Sensor Sidik Jari Dan Sensor Proximity Infrared Berbasis Arduino Uno. *Elektrika*, *16*(1), 35. https://doi.org/10.26623/elektrika.v16i1.8252

Berbasis, L., Internet, I., & Hamid, M. I. (2024). *Arus Jurnal Sains dan Teknologi (AJST) Perancangan Sistem Smart Home dengan Pengontrolan Empat Beban*. *2*(2).

Hasan, Z., Pamungkas, B., Mahdavikia, M. M., & Jaya, P. N. H. (2024). Faktor Penyebab dan Upaya Penanggulangan Tindak Pidana Pencurian Kendaraan Bermotor. *JALAKOTEK: Journal of Accounting Law Communication and Technology*, *1*(2), 316–323. https://doi.org/10.57235/jalakotek.v1i2.2369

Husniyah, F., Ulum, M., Aji Wibisono, K., & Alfita, R. (2021). Rancang Bangun Sistem Pengaman Pintu Menggunakan RFID dan Fingerprint. *Jurnal FORTECH*, *2*(1), 1–8. https://doi.org/10.32492/fortech.v2i1.232

Jardian, J., & Owen, M. (2024). *Perancangan Smart Door Berbasis ESP32-WROVER dengan Sistem Notifikasi Melalui Aplikasi Blynk*. https://doi.org/10.37253/telcomatics.v9i2.10096

Made Budiada, I., Bagus, I., Purnama, I., Alit, P., Santiary, W., Swardika, K., Nyoman, I., Wardana, K., Budiada, I. M., Purnama, I. B. I., Santiary, P. A. W., Swardika, I. K., & Wardana, I. N. K. (2024). Design and implementation of IoT-based motorcycle keyless ignition and starter using RFID and Blynk. *Matrix: Jurnal Manajemen Teknologi Dan Informatika*, *14*(3), 119–127.

Roza, Y., Musliadi, K., & Pernando, Y. (2024). Presence Prototype Model Design Using RFID RC522. *Scientific Literacy Innovation and Technology Journal*, *1*(02), 26–31.

Safitri, L., & Prasetyo, G. (2022). Rancang Bangun Alat Pengendali Pompa dan Pemantauan Batas Minimum Larutan Hara pada Metode Aeroponik Menggunakan Mikrokontroler ESP32. *Prosiding Seminar Nasional Ilmu ...*, 31–37.

Saifudin, A., Febriansyah, A., Adianto, E. B. N., Putra, E. M., & Yulianti, Y. (2023). Pengembangan Sistem Informasi Toko Online Menggunakan Metode Prototyping untuk Penjualan Produk Furniture. *Jurnal Teknologi Sistem Informasi Dan Aplikasi*, *6*(2), 173–179. https://doi.org/10.32493/jtsi.v6i2.26324

Salim, D. N., Pujisusilo, N. A., & Manik, S. P. (2021). Sistem Keamanan Smart Door Lock Menggunakan E-KTP (Elektroknik Kartu Tanda Penduduk) Berbasis Internet of Things (IoT). *Go Infotech: Jurnal Ilmiah STMIK AUB*, *27*(2), 196–206. https://doi.org/10.36309/goi.v27i2.157

Saputra, J., Rizaldi, R., Ali, S.-, Mellyssa, W., & Usmardi, U. (2020). Sistem Pengaman Pintu Menggunakan Sidik Jari Dan Android. *VOCATECH: Vocational Education and Technology Journal*, *2*(1), 33–40. https://doi.org/10.38038/vocatech.v2i1.32

Sari, I. V., Darmayanti, D. R., Widiasari, C., Indani, W., & Sitopu, M. W. (2024). Sistem Otomatis Penyiraman Dan Pemupukan Tanaman Tin Menggunakan Mikrokontroler Esp32. *Jurnal Informatika Dan Teknik Elektro Terapan*, *12*(3). https://doi.org/10.23960/jitet.v12i3.4564

Somantri, S., Insany, G. P., & Ryansyah, R. (2024). Pengembangan Sistem Keamanan Kendaraan Bermotor Menggunakan Teknologi Fingerprint dengan Metode Prototype Berbasis Internet Of Things. *G-Tech: Jurnal Teknologi Terapan*, *8*(1), 593–602. https://doi.org/10.33379/gtech.v8i1.3879

Wibowo, A. D., & Susanto, E. S. (2024). Perancangan Purwarupa Smart Door Lock Berbasis Internet of Things. *Jurnal Elektro & Informatika Swadharma (Jeis)*, *4*(2), 1–8.

LAMPIRAN

Lampiran 1 Program *Smart Padlock*

Program *Smart Padlock* menggunakan Arduino IDE dengan bahasa pemrograman C++. Berikut kode program utama *Smart Padlock* :

```
#define BLYNK_TEMPLATE_ID "TMPL6EOgKBnw4"
#define BLYNK_TEMPLATE_NAME "Smart Padlock"
#define BLYNK_AUTH_TOKEN "BS-LtXu0E_btbq_oYIamuDFLIPNCsg9Z"

#include <WiFi.h>
#include <BlynkSimpleEsp32.h>

#include <Adafruit_Fingerprint.h>
#include <HardwareSerial.h>
#include <SPI.h>
#include <MFRC522.h>
#include <Preferences.h>
Preferences preferences;

// ========== Konfigurasi PIN ==========
#define RST_PIN 22
#define SS_PIN 21
#define RELAY_PIN 5
#define BUZZER_PIN 4

// ========== RFID ==========
#define MAX_CARDS 20
MFRC522 mfrc522(SS_PIN, RST_PIN);
byte registeredUIDs[MAX_CARDS][10];
byte uidSizes[MAX_CARDS];
int registeredCount = 0;

// ========== Fingerprint ==========
HardwareSerial mySerial(2);
Adafruit_Fingerprint finger(&mySerial);

// ========== State ==========
bool relayState = false;

char auth[] = "BS-LtXu0E_btbq_oYIamuDFLIPNCsg9Z";
char ssid[] = "SmartPadlock";
char pass[] = "11111111";

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

  pinMode(RELAY_PIN, OUTPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  digitalWrite(RELAY_PIN, LOW);
  digitalWrite(BUZZER_PIN, LOW);

  mySerial.begin(57600, SERIAL_8N1, 16, 17);
```

```
  finger.begin(57600);
  if (!finger.verifyPassword()) {
    Serial.println("Fingerprint tidak terdeteksi!");
    while (true) delay(100);
  }

  SPI.begin();
  mfrc522.PCD_Init();
  preferences.begin("rfid", false);
  loadRFIDData();

  Serial.println("Smart Padlock Siap!");

  Blynk.begin(auth, ssid, pass);
}

// ====================== Loop ======================
void loop() {
  Blynk.run();
  if (checkFingerprint()) {
    toggleRelay();
    delay(2000);
  } else if (checkRFID()) {
    toggleRelay();
    delay(2000);
  }
}

// ==================== Fingerprint ====================
bool checkFingerprint() {
  if (finger.getImage() != FINGERPRINT_OK) return false;
  if (finger.image2Tz() != FINGERPRINT_OK) return false;
  if (finger.fingerFastSearch() != FINGERPRINT_OK) {
    Serial.println("❌ Sidik jari tidak ditemukan.");
    beepGagal();
    return false;
  }

  Serial.print("✔ Sidik jari cocok. ID: ");
  Serial.println(finger.fingerID);
  return true;
}

// ==================== RFID ====================
bool checkRFID() {
  if (!mfrc522.PICC_IsNewCardPresent()) return false;
  if (!mfrc522.PICC_ReadCardSerial()) return false;

  if (isCardRegistered(mfrc522.uid.uidByte, mfrc522.uid.size)) {
    Serial.print("✔ Kartu RFID cocok: ");
    for (byte i = 0; i < mfrc522.uid.size; i++) {
      Serial.print(mfrc522.uid.uidByte[i], HEX); Serial.print("
");
    }
    Serial.println();
    mfrc522.PICC_HaltA();
```

```
    return true;
  } else {
    Serial.println("❌ Kartu RFID tidak terdaftar.");
    beepGagal();
    mfrc522.PICC_HaltA();
    return false;
  }
}

// ==================== Relay dan Buzzer ====================
void toggleRelay() {
  relayState = !relayState;
  digitalWrite(RELAY_PIN, relayState ? HIGH : LOW);

  // Update ke Blynk
  Blynk.virtualWrite(V1, relayState ? "TERKUNCI" : "TERBUKA");

  if (relayState) {
    Serial.println("🔒 Solenoid MENGUNCI");
    Blynk.logEvent("solenoid_lock", "Smart Padlock terkunci!");
  } else {
    Serial.println("🔓 Solenoid TERBUKA");
    Blynk.logEvent("solenoid_unlock", "Smart Padlock terbuka!");
  }

  beepSukses();
}

void beepSukses() {
  for (int i = 0; i < 2; i++) {
    digitalWrite(BUZZER_PIN, HIGH);
    delay(100);
    digitalWrite(BUZZER_PIN, LOW);
    delay(100);
  }
}

void beepGagal() {
  digitalWrite(BUZZER_PIN, HIGH);
  delay(700); // bunyi panjang
  digitalWrite(BUZZER_PIN, LOW);
}

// ==================== RFID Helpers ====================
bool compareUID(byte *uid1, byte size1, byte *uid2, byte size2)
{
  if (size1 != size2) return false;
  for (byte i = 0; i < size1; i++) {
    if (uid1[i] != uid2[i]) return false;
  }
  return true;
}

bool isCardRegistered(byte *uid, byte size) {
  for (int i = 0; i < registeredCount; i++) {
```

```
    if (compareUID(uid, size, registeredUIDs[i], uidSizes[i]))
return true;
  }
  return false;
}

void loadRFIDData() {
  registeredCount = preferences.getInt("count", 0);
  for (int i = 0; i < registeredCount; i++) {
    String key = "uid" + String(i);
    uidSizes[i] = preferences.getUChar(("len" +
String(i)).c_str(), 0);
    preferences.getBytes(key.c_str(), registeredUIDs[i],
uidSizes[i]);
  }
}
```

Lampiran 2 Program Registrasi Sidik Jari Dan RFID

Sebelum *Smart Padlock* menggunakan program utama, pastikan sudah mendaftarkan sidik jari dan kartu RFID. Untuk mendaftarkan sidik jari dan RFID menggunakan kode program yang berbeda. Berikut kode program pendaftaran sidik jari dan kartu RFID :

```
#include <Adafruit_Fingerprint.h>
#include <HardwareSerial.h>
#include <SPI.h>
#include <MFRC522.h>
#include <Preferences.h>
Preferences preferences;

// ------------------ PIN KONFIGURASI -----------------
#define RST_PIN 22
#define SS_PIN 21
#define RELAY_PIN 5     // D5
#define BUZZER_PIN 4    // D4

// ------------------ RFID KONFIGURASI -----------------
#define MAX_CARDS 20
MFRC522 mfrc522(SS_PIN, RST_PIN);
byte registeredUIDs[MAX_CARDS][10];
byte uidSizes[MAX_CARDS];
int registeredCount = 0;

// ------------------ FINGERPRINT -----------------
HardwareSerial mySerial(2);  // RX = GPIO16, TX = GPIO17
Adafruit_Fingerprint finger(&mySerial);

// ------------------ RELAY STATE -----------------
bool relayState = false;

// ------------------ SETUP -----------------
void setup() {
  Serial.begin(115200);
```

```
  delay(1000);

  // Inisialisasi pin
  pinMode(RELAY_PIN, OUTPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  digitalWrite(RELAY_PIN, LOW);  // Awal relay OFF
  digitalWrite(BUZZER_PIN, LOW);

  // Fingerprint
  Serial.println("Inisialisasi Fingerprint...");
  mySerial.begin(57600, SERIAL_8N1, 16, 17);
  finger.begin(57600);
  if (finger.verifyPassword()) {
    Serial.println("Fingerprint OK");
  } else {
    Serial.println("Gagal inisialisasi fingerprint");
    while (1) delay(100);
  }

  // RFID
  Serial.println("Inisialisasi RFID...");
  SPI.begin();
  mfrc522.PCD_Init();
  Serial.println("RFID Siap");

  // Preferences (EEPROM)
  preferences.begin("rfid", false);
  loadRFIDData();

  printMenu();
}

// ------------------ LOOP ------------------
void loop() {
  if (Serial.available()) {
    String pilihan = readSerialLine();

    if (pilihan == "1") enrollFinger();
    else if (pilihan == "2") deleteFingerprint();
    else if (pilihan == "3") scanFingerprint();
    else if (pilihan == "4") daftarRFID();
    else if (pilihan == "5") hapusRFID();
    else if (pilihan == "6") cekRFID();
    else Serial.println("Pilihan tidak valid.");

    printMenu();
  }
}

// ------------------ UTIL ------------------
String readSerialLine() {
  String input = "";
  while (true) {
    if (Serial.available() > 0) {
      char c = Serial.read();
      if (c == '\n' || c == '\r') {
```

```
      if (input.length() > 0) break;
    } else {
      input += c;
      Serial.print(c);
    }
  }
}
Serial.println();
return input;
}

void printMenu() {
  Serial.println("\n=== MENU UTAMA ===");
  Serial.println("1. Daftar Sidik Jari");
  Serial.println("2. Hapus Sidik Jari");
  Serial.println("3. Cek Sidik Jari");
  Serial.println("4. Daftar Kartu RFID");
  Serial.println("5. Hapus Kartu RFID");
  Serial.println("6. Cek Kartu RFID");
  Serial.print("Pilih opsi: ");
}

// ----------------- FINGERPRINT -----------------
void enrollFinger() {
  Serial.println("Masukkan ID sidik jari (0-127):");
  String idStr = readSerialLine();
  int id = idStr.toInt();
  if (id < 0 || id > 127) {
    Serial.println("ID tidak valid");
    return;
  }

  Serial.println("Letakkan jari untuk scan pertama...");
  while (finger.getImage() != FINGERPRINT_OK) delay(100);
  if (finger.image2Tz(1) != FINGERPRINT_OK) {
    Serial.println("Gagal proses gambar ke template");
    return;
  }

  Serial.println("Lepas dan letakkan lagi jari...");
  delay(2000);
  while (finger.getImage() != FINGERPRINT_OK) delay(100);
  if (finger.image2Tz(2) != FINGERPRINT_OK) {
    Serial.println("Gagal proses gambar ke template 2");
    return;
  }

  if (finger.createModel() != FINGERPRINT_OK ||
finger.storeModel(id) != FINGERPRINT_OK) {
    Serial.println("Gagal menyimpan sidik jari");
  } else {
    Serial.println("Sidik jari berhasil didaftarkan");
  }
}

void deleteFingerprint() {
```

```
  Serial.println("Masukkan ID sidik jari yang ingin dihapus:");
  String idStr = readSerialLine();
  int id = idStr.toInt();
  if (finger.deleteModel(id) == FINGERPRINT_OK) {
    Serial.println("Sidik jari dihapus.");
  } else {
    Serial.println("Gagal menghapus.");
  }
}

void scanFingerprint() {
  Serial.println("Letakkan jari Anda...");
  int p = -1;
  unsigned long startTime = millis();

  // Mulai proses pencarian sidik jari
  while (millis() - startTime < 3000) {
    p = finger.getImage();
    if (p == FINGERPRINT_OK) break;
    delay(30);
  }

  unsigned long endTime = millis();
  float responseTime = (endTime - startTime) / 1000.0;

  if (p != FINGERPRINT_OK) {
    Serial.print("Gagal membaca gambar sidik jari. Response
time: ");
    Serial.print(responseTime, 3);
    Serial.println(" detik");
    return;
  }

  if (finger.image2Tz() != FINGERPRINT_OK ||
finger.fingerFastSearch() != FINGERPRINT_OK) {
    Serial.print("Sidik jari tidak ditemukan. Response time: ");
    Serial.print(responseTime, 3);
    Serial.println(" detik");
  } else {
    Serial.print("Sidik jari ditemukan. ID: ");
    Serial.print(finger.fingerID);
    Serial.print(" | Response time: ");
    Serial.print(responseTime, 3);
    Serial.println(" detik");
    toggleRelay();
  }
}

// ------------------ RFID ------------------
void daftarRFID() {
  Serial.println("Tempelkan kartu...");
  while (!mfrc522.PICC_IsNewCardPresent() ||
!mfrc522.PICC_ReadCardSerial()) delay(100);
  if (addCard(mfrc522.uid.uidByte, mfrc522.uid.size)) {
    Serial.println("Kartu berhasil ditambahkan.");
  } else {
```

```
    Serial.println("Kartu sudah ada atau kapasitas penuh.");
  }
  mfrc522.PICC_HaltA();
}

void hapusRFID() {
  Serial.println("Tempelkan kartu untuk dihapus...");
  while (!mfrc522.PICC_IsNewCardPresent() ||
!mfrc522.PICC_ReadCardSerial()) delay(100);
  if (removeCard(mfrc522.uid.uidByte, mfrc522.uid.size)) {
    Serial.println("Kartu berhasil dihapus.");
  } else {
    Serial.println("Kartu tidak ditemukan.");
  }
  mfrc522.PICC_HaltA();
}

void cekRFID() {
  Serial.println("Tempelkan kartu untuk dicek...");
  unsigned long startTime = millis();
  bool found = false;

  while (millis() - startTime < 3000) {
    if (mfrc522.PICC_IsNewCardPresent() &&
mfrc522.PICC_ReadCardSerial()) {
      found = true;
      break;
    }
    delay(30);
  }

  unsigned long endTime = millis();
  float responseTime = (endTime - startTime) / 1000.0;

  if (!found) {
    Serial.print("Kartu tidak terdeteksi. Response time: ");
    Serial.print(responseTime, 3);
    Serial.println(" detik");
    return;
  }

  // Tampilkan UID kartu
  Serial.print("UID: ");
  for (byte i = 0; i < mfrc522.uid.size; i++) {
    Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? "0" : "");
    Serial.print(mfrc522.uid.uidByte[i], HEX);
    if (i < mfrc522.uid.size - 1) Serial.print(":");
  }
  Serial.println();

  // Cek apakah UID sudah terdaftar
  if (isCardRegistered(mfrc522.uid.uidByte, mfrc522.uid.size)) {
    Serial.print("Kartu DITEMUKAN. Response time: ");
    Serial.print(responseTime, 3);
    Serial.println(" detik");
    toggleRelay();
```

```
  } else {
    Serial.print("Kartu tidak terdaftar. Response time: ");
    Serial.print(responseTime, 3);
    Serial.println(" detik");
  }

  mfrc522.PICC_HaltA();
}


// ------------------ EEPROM / Preferences ------------------
void saveRFIDData() {
  preferences.putInt("count", registeredCount);
  for (int i = 0; i < registeredCount; i++) {
    String key = "uid" + String(i);
    preferences.putBytes(key.c_str(), registeredUIDs[i],
uidSizes[i]);
    preferences.putUChar(("len" + String(i)).c_str(),
uidSizes[i]);
  }
}

void loadRFIDData() {
  registeredCount = preferences.getInt("count", 0);
  for (int i = 0; i < registeredCount; i++) {
    String key = "uid" + String(i);
    uidSizes[i] = preferences.getUChar(("len" +
String(i)).c_str(), 0);
    preferences.getBytes(key.c_str(), registeredUIDs[i],
uidSizes[i]);
  }
}

// ------------------ LOGIKA AKSES ------------------
bool compareUID(byte *uid1, byte size1, byte *uid2, byte size2)
{
  if (size1 != size2) return false;
  for (byte i = 0; i < size1; i++) {
    if (uid1[i] != uid2[i]) return false;
  }
  return true;
}

bool isCardRegistered(byte *uid, byte size) {
  for (int i = 0; i < registeredCount; i++) {
    if (compareUID(uid, size, registeredUIDs[i], uidSizes[i]))
return true;
  }
  return false;
}

bool addCard(byte *uid, byte size) {
  if (registeredCount >= MAX_CARDS || isCardRegistered(uid,
size)) return false;
  for (byte i = 0; i < size; i++) {
    registeredUIDs[registeredCount][i] = uid[i];
```

```
  }
  uidSizes[registeredCount] = size;
  registeredCount++;
  saveRFIDData();
  return true;
}

bool removeCard(byte *uid, byte size) {
  for (int i = 0; i < registeredCount; i++) {
    if (compareUID(uid, size, registeredUIDs[i], uidSizes[i])) {
      for (int j = i; j < registeredCount - 1; j++) {
        memcpy(registeredUIDs[j], registeredUIDs[j + 1],
sizeof(registeredUIDs[j]));
        uidSizes[j] = uidSizes[j + 1];
      }
      registeredCount--;
      saveRFIDData();
      return true;
    }
  }
  return false;
}

// ----------------- RELAY DAN BUZZER -----------------
void toggleRelay() {
  relayState = !relayState;
  digitalWrite(RELAY_PIN, relayState ? HIGH : LOW);
  beepBuzzer(2);
}

void beepBuzzer(int count) {
  for (int i = 0; i < count; i++) {
    digitalWrite(BUZZER_PIN, HIGH);
    delay(100);
    digitalWrite(BUZZER_PIN, LOW);
    delay(100);
  }
}
```