

LAMPIRAN

```
19 #include <Adafruit_Fingerprint.h>
20
21 HardwareSerial fingerSerial(2);
22 HardwareSerial gsmSerial(1);
23 Adafruit_Fingerprint finger = Adafruit_Fingerprint(&fingerSerial);
24
25 #define RELAY_PIN 26
26 #define BUZZER_PIN 27
27
28 // Nomor HP pemilik untuk notifikasi otomatis
29 #define NOMOR_HP "+6282282297995"
30
31 // State sistem
32 bool motorAktif = false;
33 bool sistemTerkunci = false;
34 unsigned long lastSMSCheck = 0;
35 #define SMS_CHECK_INTERVAL 5000
36
37 // =====
38 // SETUP
39 // =====
40 void setup() {
41   Serial.begin(115200);
42   delay(500);
43
44   pinMode(RELAY_PIN, OUTPUT);
45   pinMode(BUZZER_PIN, OUTPUT);
46   digitalWrite(RELAY_PIN, LOW);
47   digitalWrite(BUZZER_PIN, LOW);
48
49   fingerSerial.begin(57600, SERIAL_8N1, 16, 17);
50
51   finger.begin(57600);
52   delay(100);
53
54   gsmSerial.begin(9600, SERIAL_8N1, 18, 19);
55   delay(1000);
56
57   Serial.println("=====");
58   Serial.println(" SISTEM KEAMANAN SEPEDA MOTOR");
59   Serial.println(" MARIANI - 2208100055");
60   Serial.println("=====");
61
62   if (finger.verifyPassword()) {
63     Serial.println("[OK] Fingerprint siap.");
64   } else {
65     Serial.println("[ERROR] Fingerprint tidak ditemukan!");
66     buzzerError();
67     while (true);
68   }
69
70   initGSM();
71
72   Serial.println("[SIAGA] Sistem aktif.");
73   Serial.println("Kirim 'a' di Serial Monitor untuk tambah sidik jari.");
74 }
75 // =====
76 // LOOP
77 // =====
78 void loop() {
79   // Cek perintah Serial Monitor
80   if (Serial.available()) {
81     char input = Serial.read();
82     if (input == 'a') tambahSidikJariSerial();
```

```

82     }
83
84     // Cek SMS masuk setiap 5 detik
85     if (millis() - lastSMSCheck >= SMS_CHECK_INTERVAL) {
86         lastSMSCheck = millis();
87         cekSMSMasuk();
88     }
89
90     // Scan sidik jari hanya kalau motor mati dan tidak terkunci
91     if (!motorAktif && !sistemTerkunci) {
92         int result = scanSidikJari();
93
94         if (result > 0) {
95             Serial.print("[OK] Dikenali! ID #"); Serial.println(result);
96             aktifkanMotor();
97             kirimSMS("Motor dinyalakan. Akses diberikan oleh sidik jari ID #" + String(result));
98         } else if (result == -1) {
99             Serial.println("[DITOLAK] Sidik jari tidak terdaftar!");
100             buzzerAlert();
101             kirimSMS("PERINGATAN! Ada percobaan akses tidak sah pada motor Anda!");
102             delay(3000);
103         }
104     }
105 }
106
107 delay(100);
108 }
109
110 // =====
111 //  SCAN SIDIK JARI
112 // =====
113 int scanSidikJari() {
114     uint8_t p = finger.getImage();
115     if (p == FINGERPRINT_NOFINGER) return 0;
116     if (p != FINGERPRINT_OK) return 0;
117
118     p = finger.image2Tz();
119     if (p != FINGERPRINT_OK) return 0;
120
121     p = finger.fingerSearch();
122     if (p == FINGERPRINT_OK) return finger.fingerID;
123     if (p == FINGERPRINT_NOTFOUND) return -1;
124     return 0;
125 }
126
127 // =====
128 //  KONTROL MOTOR
129 // =====
130 void aktifkanMotor() {
131     digitalWrite(RELAY_PIN, HIGH);
132     motorAktif = true;
133     sistemTerkunci = false;
134     buzzerOK();
135     Serial.println("[RELAY] ON - Pengapian tersambung.");
136 }
137
138 void matikanMotor() {
139     digitalWrite(RELAY_PIN, LOW);
140     motorAktif = false;
141     Serial.println("[RELAY] OFF - Pengapian diputus.");
142 }
143
144 // =====
145 //  CEK SMS MASUK

```

```

177 // Hapus SMS dulu sebelum eksekusi
178 hapusSMS(indexSMS);
179
180 // Eksekusi perintah
181 if (responseUpper.indexOf("MATIKAN") >= 0) {
182     Serial.println("[SMS CMD] MATIKAN");
183     matikanMotor();
184     sistemTerkunci = true;
185     buzzerAlert();
186     kirimSMSke(nomorPengirim, "Perintah diterima. Pengapian motor telah DIMATIKAN dari jarak jauh.");
187 }
188 else if (responseUpper.indexOf("NYALAKAN") >= 0) {
189     Serial.println("[SMS CMD] NYALAKAN");
190     aktifkanMotor();
191     kirimSMSke(nomorPengirim, "Perintah diterima. Pengapian motor telah DINYALAKAN dari jarak jauh.");
192 }
193 else if (responseUpper.indexOf("STATUS") >= 0) {
194     Serial.println("[SMS CMD] STATUS");
195     String statusMotor = motorAktif ? "MENYALA" : "MATI";
196     String statusKunci = sistemTerkunci ? "TERKUNCI (via SMS)" : "NORMAL";
197     kirimSMSke(nomorPengirim, "STATUS MOTOR: " + statusMotor + " | SISTEM: " + statusKunci);
198 }
199 else if (responseUpper.indexOf("DAFTARJARI") >= 0) {
200     Serial.println("[SMS CMD] DAFTARJARI");
201     enrollViaSMS(nomorPengirim);
202 }
203 else {
204     // Perintah tidak dikenal
205     kirimSMSke(nomorPengirim, "Perintah tidak dikenal. Perintah tersedia: MATIKAN, NYALAKAN, STATUS, DAFTARJARI");
206 }
207 }
208
209 // CEK SMS MASUK
210 // =====
211 void cekSMSMasuk() {
212     gsmSerial.println("AT+CMGL=\"REC UNREAD\"");
213     delay(2000);
214
215     String response = "";
216     while (gsmSerial.available()) {
217         response += (char)gsmSerial.read();
218     }
219
220     if (response.indexOf("+CMGL:") < 0) return;
221
222     // Ambil index SMS
223     int idxCMGL = response.indexOf("+CMGL:");
224     int idxKoma = response.indexOf(",", idxCMGL);
225     String indexSMS = response.substring(idxCMGL + 7, idxKoma);
226     indexSMS.trim();
227
228     // Ambil nomor pengirim
229     // Format: +CMGL: 1,"REC UNREAD", "+628xxx", "date"
230     int idxNomor1 = response.indexOf("\",\"", idxCMGL);
231     int idxNomor2 = response.indexOf("\"", idxNomor1 + 3);
232     String nomorPengirim = response.substring(idxNomor1 + 3, idxNomor2);
233     nomorPengirim.trim();
234
235     Serial.println("[GSM] SMS dari: " + nomorPengirim);
236     Serial.println("[GSM] Index: #" + indexSMS);
237
238     String responseUpper = response;
239     responseUpper.toUpperCase();
240 }

```

```

209 // =====
210 // HAPUS SMS
211 // =====
212 void hapusSMS(String index) {
213     gsmSerial.println("AT+CMGD=" + index + ",0");
214     delay(1000);
215     Serial.println("[GSM] SMS #" + index + " dihapus.");
216 }
217
218 // =====
219 // ENROLL VIA SMS
220 // =====
221 void enrollViaSMS(String nomorPengirim) {
222     finger.getTemplateCount();
223     uint8_t newID = finger.templateCount + 1;
224
225     if (newID > 127) {
226         kirimSMSke(nomorPengirim, "Gagal! Memori sidik jari penuh (maks 127).");
227         return;
228     }
229
230     Serial.println("[ENROLL] Mode daftar sidik jari aktif.");
231     kirimSMSke(nomorPengirim, "Mode daftar sidik jari aktif. Tempelkan jari ke sensor sekarang! ID #" + String(newID));
232     buzzerEnrollMode();
233
234     int p = -1;
235     unsigned long timeout = millis();
236
237     // Scan pertama (timeout 15 detik)
238     Serial.println("Menunggu jari pertama...");
239     while (p != FINGERPRINT_OK) {
240         // Hapus SMS dulu sebelum eksekusi
241         hapusSMS(indexSMS);
242
243         // Eksekusi perintah
244         if (responseUpper.indexOf("MATIKAN") >= 0) {
245             Serial.println("[SMS CMD] MATIKAN");
246             matikanMotor();
247             sistemTerkunci = true;
248             buzzerAlert();
249             kirimSMSke(nomorPengirim, "Perintah diterima. Pengapian motor telah DIMATIKAN dari jarak jauh.");
250         } else if (responseUpper.indexOf("NYALAKAN") >= 0) {
251             Serial.println("[SMS CMD] NYALAKAN");
252             aktifkanMotor();
253             kirimSMSke(nomorPengirim, "Perintah diterima. Pengapian motor telah DINYALAKAN dari jarak jauh.");
254         } else if (responseUpper.indexOf("STATUS") >= 0) {
255             Serial.println("[SMS CMD] STATUS");
256             String statusMotor = motorAktif ? "MENYALA" : "MATI";
257             String statusKunci = sistemTerkunci ? "TERKUNCI (via SMS)" : "NORMAL";
258             kirimSMSke(nomorPengirim, "STATUS MOTOR: " + statusMotor + " | SISTEM: " + statusKunci);
259         } else if (responseUpper.indexOf("DAFTARJARI") >= 0) {
260             Serial.println("[SMS CMD] DAFTARJARI");
261             enrollViaSMS(nomorPengirim);
262         } else {
263             // Perintah tidak dikenal
264             kirimSMSke(nomorPengirim, "Perintah tidak dikenal. Perintah tersedia: MATIKAN, NYALAKAN, STATUS, DAFTARJARI");
265         }
266     }
267 }
268

```

```

271 timeout = millis();
272 Serial.println("Menunggu jari kedua...");
273 while (p != FINGERPRINT_OK) {
274     if (millis() - timeout > 15000) {
275         kirimSMSke(nomorPengirim, "Gagal! Waktu habis di scan kedua. Kirim DAFTARJARI lagi untuk coba ulang.");
276         return;
277     }
278     p = finger.getImage();
279     if (p == FINGERPRINT_OK) {
280         Serial.println("[OK] Jari kedua terdeteksi.");
281         buzzerEnrollTunggu();
282     }
283 }
284
285 p = finger.image2Tz(2);
286 if (p != FINGERPRINT_OK) {
287     kirimSMSke(nomorPengirim, "Gagal proses gambar jari kedua. Coba lagi.");
288     return;
289 }
290
291 p = finger.createModel();
292 if (p == FINGERPRINT_ENROLLMISMATCH) {
293     buzzerError();
294     kirimSMSke(nomorPengirim, "Gagal! Jari tidak cocok antara scan 1 dan 2. Kirim DAFTARJARI lagi.");
295     return;
296 }
297 if (p != FINGERPRINT_OK) {
298     kirimSMSke(nomorPengirim, "Gagal buat model sidik jari. Coba lagi.");
299     return;
300 }
301
302 if (millis() - timeout > 15000) {
303     kirimSMSke(nomorPengirim, "Gagal! Waktu habis. Kirim DAFTARJARI lagi untuk coba ulang.");
304     return;
305 }
306
307 p = finger.getImage();
308 if (p == FINGERPRINT_OK) {
309     Serial.println("[OK] Jari pertama terdeteksi.");
310     buzzerEnrollTunggu();
311 }
312
313 p = finger.image2Tz(1);
314 if (p != FINGERPRINT_OK) {
315     kirimSMSke(nomorPengirim, "Gagal proses gambar jari. Coba lagi.");
316     return;
317 }
318
319 // Jeda angkat jari
320 for (int i = 0; i < 2; i++) {
321     digitalWrite(BUZZER_PIN, HIGH); delay(150);
322     digitalWrite(BUZZER_PIN, LOW); delay(150);
323 }
324
325 kirimSMSke(nomorPengirim, "Jari pertama OK! Angkat jari lalu tempelkan jari yang SAMA sekali lagi.");
326 Serial.println("Angkat jari...");
327 delay(2000);
328 p = 0;
329 while (p != FINGERPRINT_NOFINGER) p = finger.getImage();
330
331 // Scan kedua (timeout 15 detik)
332 p = -1;

```

```

330     p = finger.getImage();
331     if (p == FINGERPRINT_OK) Serial.println("[OK] Gambar pertama diambil.");
332     else if (p == FINGERPRINT_NOFINGER) Serial.print(".");
333     else { Serial.println("[ERROR]"); return; }
334 }
335
336 p = finger.image2Tz(1);
337 if (p != FINGERPRINT_OK) { Serial.println("[ERROR] Gagal proses gambar 1."); return; }
338
339 Serial.println("Angkat jari...");
340 delay(2000);
341 p = 0;
342 while (p != FINGERPRINT_NOFINGER) p = finger.getImage();
343
344 Serial.println("Tempelkan jari yang sama lagi...");
345 p = -1;
346 while (p != FINGERPRINT_OK) {
347     p = finger.getImage();
348     if (p == FINGERPRINT_OK) Serial.println("[OK] Gambar kedua diambil.");
349     else if (p == FINGERPRINT_NOFINGER) Serial.print(".");
350     else { Serial.println("[ERROR]"); return; }
351 }
352
353 p = finger.image2Tz(2);
354 if (p != FINGERPRINT_OK) { Serial.println("[ERROR] Gagal proses gambar 2."); return; }
355
356 p = finger.createModel();
357 if (p == FINGERPRINT_ENROLLMISMATCH) { Serial.println("[ERROR] Jari tidak cocok, ulangi."); return; }
358 if (p != FINGERPRINT_OK) { Serial.println("[ERROR] Gagal buat model."); return; }
359 p = finger.storeModel(newID);
360 if (p == FINGERPRINT_OK) {
361     Serial.print("[OK] Tersimpan di ID #"); Serial.println(newID);
362     buzzerOK();
363     kirimSMSke(nomorPengirim, "Sidik jari berhasil didaftarkan! ID # " + String(newID) + ". Total: " + String(newID) + " ja
364 } else {
365     buzzerError();
366     kirimSMSke(nomorPengirim, "Gagal menyimpan sidik jari. Coba lagi.");
367 }
368 }
369 }
370
371 // =====
372 // ENROLL VIA SERIAL MONITOR
373 // =====
374 void tambahSidikJariSerial() {
375     finger.getTemplateCount();
376     uint8_t newID = finger.templateCount + 1;
377
378     if (newID > 127) {
379         Serial.println("[ERROR] Memori penuh!");
380         return;
381     }
382
383     Serial.print("\n[ENROLL] Mendaftarkan ID #"); Serial.println(newID);
384
385     int p = -1;
386     Serial.println("Tempelkan jari sekarang...");
387     while (p != FINGERPRINT_OK) {

```

```

388     gsmSerial.println("AT+CMGF=1");
389     delay(500);
390     gsmSerial.print("AT+CMGS=\"");
391     gsmSerial.print(nomor);
392     gsmSerial.println("\");
393     delay(1000);
394     gsmSerial.print(pesan);
395     delay(500);
396     gsmSerial.write(26);
397     delay(3000);
398     Serial.println("[OK] SMS terkirim.");
399 }
400
401 // =====
402 // BUZZER
403 // =====
404 void buzzerOK() {
405     // 2 pendek = akses OK / enroll berhasil
406     for (int i = 0; i < 2; i++) {
407         digitalWrite(BUZZER_PIN, HIGH); delay(150);
408         digitalWrite(BUZZER_PIN, LOW); delay(150);
409     }
410 }
411
412 void buzzerAlert() {
413     // 5 panjang = akses ditolak / motor dimatikan paksa
414     for (int i = 0; i < 5; i++) {
415         digitalWrite(BUZZER_PIN, HIGH); delay(400);
416         digitalWrite(BUZZER_PIN, LOW); delay(200);
417     }
418 }
419
420 p = finger.storeModel(newID);
421 if (p == FINGERPRINT_OK) {
422     Serial.print("[OK] Tersimpan di ID #"); Serial.println(newID);
423     buzzerOK();
424 } else {
425     Serial.println("[ERROR] Gagal simpan.");
426 }
427 }
428
429 // =====
430 // GSM - INIT & KIRIM SMS
431 // =====
432 void initGSM() {
433     Serial.println("[GSM] Inisialisasi...");
434     gsmSerial.println("AT"); delay(500);
435     gsmSerial.println("AT+CMGF=1"); delay(500);
436     gsmSerial.println("AT+CSGS=\"GSM\""); delay(500);
437     gsmSerial.println("AT+CNMI=1,1,0,0,0"); delay(500);
438     gsmSerial.println("AT+CMGDA=\"DEL ALL\""); delay(1000);
439     Serial.println("[OK] GSM siap.");
440 }
441
442 void kirimSMS(String pesan) {
443     kirimSMSke(NOMOR_HP, pesan);
444 }
445
446 void kirimSMSke(String nomor, String pesan) {
447     Serial.println("[GSM] Kirim SMS ke " + nomor + ": " + pesan);

```

```

414     for (int i = 0; i < 3; i++) {
415         digitalWrite(BUZZER_PIN, HIGH); delay(400);
416         digitalWrite(BUZZER_PIN, LOW); delay(200);
417     }
418 }
419
420 void buzzerError() {
421     // 3 panjang = error sistem
422     for (int i = 0; i < 3; i++) {
423         digitalWrite(BUZZER_PIN, HIGH); delay(600);
424         digitalWrite(BUZZER_PIN, LOW); delay(300);
425     }
426 }
427
428 void buzzerEnrollMode() {
429     // 1 panjang + 3 pendek cepat = masuk mode enroll
430     digitalWrite(BUZZER_PIN, HIGH); delay(1000);
431     digitalWrite(BUZZER_PIN, LOW); delay(200);
432     for (int i = 0; i < 3; i++) {
433         digitalWrite(BUZZER_PIN, HIGH); delay(100);
434         digitalWrite(BUZZER_PIN, LOW); delay(100);
435     }
436 }
437
438 void buzzerEnrollTunggu() {
439     // 1 pendek = jari terdeteksi, lanjut
440     digitalWrite(BUZZER_PIN, HIGH); delay(200);
441     digitalWrite(BUZZER_PIN, LOW);
442 }

```