



## Rancang bangun sistem keamanan sepeda motor menggunakan *fingerprint* berbasis arduino

**Bryan Sham Rante Tandung<sup>\*1</sup>, Dasril<sup>2</sup>, Hisma Abduh<sup>3</sup>**

Email: <sup>1</sup>briantandung@gmail.com

<sup>123</sup>Teknik Informatika, Fakultas Teknik, Universitas Andi Djemma Palopo

Diterima: 25 November 2023 | Direvisi: 28 Desember 2023 | Disetujui: 16 Januari 2024

©2024 Program Studi Teknik Informatika Fakultas Teknik,  
Universitas Andi Djemma Palopo, Indonesia

### Abstrak

Penelitian ini dilatarbelakangi oleh sistem keamanan yang ada pada sepeda motor dengan menggunakan kunci masih terbilang belum aman dikarenakan sistem keamanan yang menggunakan kunci biasa masih mudah untuk dibobol kemudian penelitian ini bertujuan untuk merancang serta mengimplementasikan sistem keamanan sepeda motor menggunakan *fingerprint* berbasis arduino. Penelitian ini dilakukan pada bengkel sepeda motor, Jln.Tandipau IV, Kelurahan Sakti, Kecamatan Bua yang dilaksanakan selama 3 bulan, dengan menggunakan data kualitatif yang terdiri dari informasi berbentuk kata dan kalimat serta data non-numerik yang diperoleh melalui wawancara dan tinjauan pustaka. Berdasarkan hasil pengamatan yang dilakukan oleh peneliti *fingerprint* akan bekerja dengan baik jika sidik jari yang ditempelkan dalam keadaan kering. *fingerprint* akan merespon dengan lambat atau bahkan tidak dapat merespon jika keadaan sidik jari basah, berminyak, berdebu dan terluka. Proses respon pembacaan pada saat menempelkan sidik jari di *fingerprint* memerlukan waktu 1 – 2 detik.

**Kata kunci:** *Fingerprint*, Sepeda Motor, Arduino

***Design and build a motorcycle security system using arduino based fingerprint***

### Abstract

*This research is motivated by the existing security system in motorcycles that still considered to be not secure enough due to the use of conventional keys, which are relatively easy to be bypassed. The purpose of this research is to design and implement a motorcycle security system using a fingerprint-based Arduino. The research was conducted at a motorcycle workshop located at Jln. Tandipau IV, Sakti Village, Bua Sub-district, and lasted for 3 months. Qualitative data was collected, consisting of text-based information and non-numeric data obtained through interviews and literature review. Based on the observations made by the researcher, the fingerprint system works effectively when the fingerprint is dry. It responds slowly or may not respond at all if the fingerprint is wet, oily, dusty, or injured. The response time for fingerprint recognition when placing a fingerprint on the sensor takes 1 to 2 seconds.*

**Keywords:** *FingerPrint, Motorcycle, Arduino.*

### 1. PENDAHULUAN

Kendaraan yang paling diminati di Indonesia ialah sepeda motor, dengan alasan utama adalah harganya yang terjangkau dan efisiensi bahan bakar yang tinggi. Penggunaan sepeda motor juga dipilih karena efisiensi waktu dan kecepatannya sebagai alat transportasi. Jumlah pengguna sepeda motor di Indonesia mencapai 80 juta unit pada tahun 2016, dan dengan meningkatnya penggunaan sepeda motor di jalan raya, kota-kota besar di Indonesia kini dipenuhi oleh sepeda motor [1]. Berdasarkan pandangan dari C. W. Wijayanti Sepeda motor merupakan kendaraan yang umum dan luas digunakan di Indonesia, sangat diminati berkat beberapa keunggulan yang dimilikinya. Kelebihan yang dimiliki oleh sepeda motor antara lain kemudahan dalam pengoperasian dan bentuknya yang ramping, memudahkan penyimpanan. Selain itu, sepeda motor memungkinkan masyarakat untuk lebih

mudah melewati kemacetan, yang merupakan salah satu permasalahan di daerah padat penduduk. [2]. Namun, seiring dengan peningkatan minat terhadap sepeda motor, terjadi peningkatan tindak kriminalitas terhadap kendaraan ini. Tidak hanya itu, sistem keamanan standar pada sepeda motor yang disertakan oleh pabrikan semakin mudah untuk ditembus. Berdasarkan Data Statistik Kriminal 2022 yang dipublikasikan oleh BPS (Badan Pusat Statistik), pada tahun 2019 terdapat 23.476 kasus pencurian kendaraan bermotor, kemudian pada tahun 2020 tercatat 18.557 kasus, dan pada tahun 2021 terdapat 18.005 kasus pencurian kendaraan sepeda motor. Dari data tersebut, terlihat bahwa jumlah kasus pencurian kendaraan sepeda motor masih cukup tinggi [3].

*Biometric identification* adalah teknologi yang memanfaatkan karakteristik atau perilaku unik dari individu untuk tujuan identifikasi. Jenis identifikasi *biometrik* yang umum melibatkan sidik jari, pengenalan wajah, pengenalan suara, serta pengenalan retina atau iris mata. Teknologi identifikasi *biometrik* menjadi salah satu metode identifikasi yang paling efektif dan aman, karena data *biometrik* dari setiap individu bersifat unik, membuatnya sulit untuk dipalsukan atau ditiru oleh pihak lain. [4]. Teknologi *Biometrik* merupakan suatu sistem yang menggunakan bagian tubuh manusia untuk memastikan [5]. Teknologi ini mengexploitasi karakteristik spesifik dari bagian tubuh manusia, seperti sidik jari, mata, dan wajah, untuk melakukan identifikasi. Saat ini, teknologi yang umumnya digunakan adalah identifikasi berbasis sidik jari. Setiap individu memiliki keunikan sidik jari yang berbeda [6]. Sementara itu, perkembangan teknologi yang pesat saat ini menciptakan kemajuan yang signifikan. Dalam perkembangan tersebut, semakin banyak komponen elektronik yang dirancang untuk melakukan identifikasi identitas dengan tingkat akurasi yang tinggi. Salah satu komponen yang sedang populer saat ini adalah sensor sidik jari atau *fingerprint*. Sensor sidik jari bekerja dengan merekam gambar digital dari pola sidik jari, dan rekaman tersebut akan dijadikan sebagai template *biometrik* yang digunakan untuk mencocokkan identitas pengguna [7].

Sistem keamanan pada sepeda motor dengan menggunakan kunci masih terbilang belum aman dikarenakan sistem keamanan yang menggunakan kunci biasa masih mudah untuk dibobol, dan sistem keamanan sepeda motor yang menggunakan sistem keyless masih memiliki beberapa kekurangan yaitu biaya perbaikan sistem keyless yang terbilang mahal, keyless tidak dapat digunakan jika aki pada sepeda motor kehabisan daya dan kemudian masalah yang pada umumnya terjadi di kedua sistem keamanan tersebut ialah hilang remote atau kunci sepeda motor, dimana jika remote dan kunci sepeda motor hilang maka motor tidak dapat digunakan kecuali mengganti remote atau kunci sepeda motor, oleh karena itu peneliti menyarankan menggunakan sidik jari sebagai alat untuk menggantikan kunci pada sepeda motor.

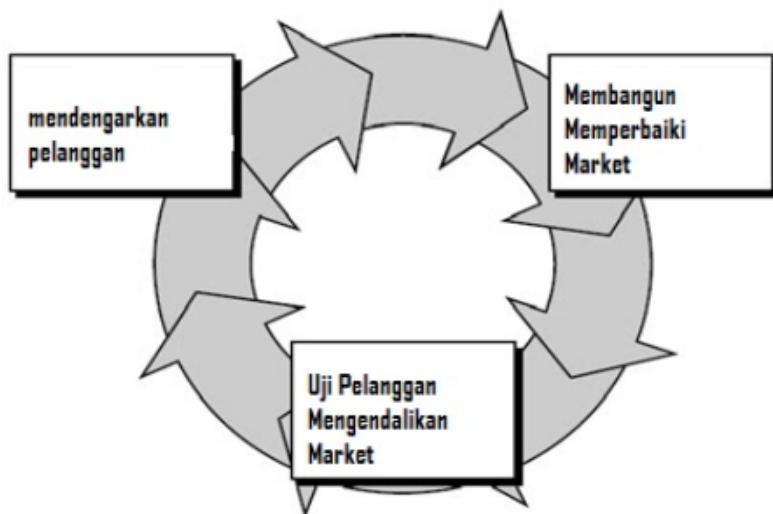
Dengan pertimbangan tersebut, peneliti memilih judul penelitian "Rancang Bangun Sistem Keamanan Sepeda Motor Menggunakan *Fingerprint* Berbasis Arduino" suatu sistem pengamanan untuk kendaraan sepeda motor yang memanfaatkan teknologi *Fingerprint* dengan dasar Arduino. Sistem ini juga dilengkapi dengan Speaker yang berfungsi memberikan peringatan saat terdeteksi tindak pencurian pada kendaraan, yang terjadi ketika sidik jari yang diidentifikasi tidak cocok. Desain ini dianggap sangat efektif untuk meningkatkan keamanan karena setiap individu memiliki sidik jari yang unik; dengan demikian, jika sidik jari yang dipindai tidak sesuai dengan yang terdaftar, sepeda motor tidak akan dapat dihidupkan. Penggunaan sidik jari juga memiliki kelebihan karena tidak memerlukan perangkat tambahan seperti kunci atau remote, sehingga pengaktifan sistem pada sepeda motor dapat dilakukan hanya dengan menggunakan kedua tangan. Tujuan dari ini juga adalah untuk mengurangi risiko kehilangan atau lupa membawa kunci atau remote ke suatu tempat.

## 2. METODE PENELITIAN

Metode pengembangan dalam rancang bangun sistem keamanan sepeda motor menggunakan *fingerprint* berbasis arduino adalah metode prototype. Metode prototype merupakan bentuk model sistem yang belum utuh menjadi sebuah hasil desain. Model ini dirancang dengan tujuan untuk memfasilitasi komunikasi antara pengembang dan calon pengguna, dengan fokus pada pendekatan "listen to customer". Oleh karena itu, dalam proses pengembangan model ini, terdapat lebih banyak interaksi komunikasi antara pengembang dan pelanggan dalam hal umpan balik terkait desainnya.

Metode prototipe melibatkan tiga tahapan, dimulai dengan tahap pertama yang disebut "*Listen to Customer*." Tahap ini mencakup proses komunikasi antara pengguna sepeda motor dan pengembang, di mana keinginan langsung dari pengguna sepeda motor dapat diimplementasikan. Kemudian, tahap kedua adalah "*Build/Revise Mock-Up*," yang melibatkan pembuatan pemodelan setengah jadi. Pada tahap ketiga, yang disebut "*Customer Test Drives Mock-Up*," dilakukan pengujian program oleh pengguna. Jika terdapat keinginan yang belum terpenuhi atau bagian yang ingin ditambahkan dari sistem yang dikembangkan, aktivitas akan kembali ke tahap awal, yaitu "*Listen to Customer*."

Menurut Roger S. Pressman, tahap awal dari model prototipe adalah komunikasi, yang bertujuan untuk mengidentifikasi masalah-masalah yang ada serta menentukan kebutuhan sistem yang harus dibangun. Langkah berikutnya adalah pembuatan model (*modelling quick design*) untuk prototipe yang akan dibuat (*construction of prototype*), diikuti oleh proses evaluasi (*feedback*) [8]. Menurut pandangan Sommerville, prototipe adalah versi awal perangkat lunak yang digunakan untuk memperlihatkan konsep, menguji berbagai opsi desain, dan mengungkap lebih banyak permasalahan serta solusinya [9]. Pendapat dari Sujono, Mayasari, dan Koloniawan (2019) menyatakan bahwa prototipe adalah alat yang memberikan gambaran kepada pembuat atau pengguna tentang bagaimana sistem berfungsi dalam bentuk lengkapnya, dan proses untuk menciptakan prototipe disebut prototyping [10].

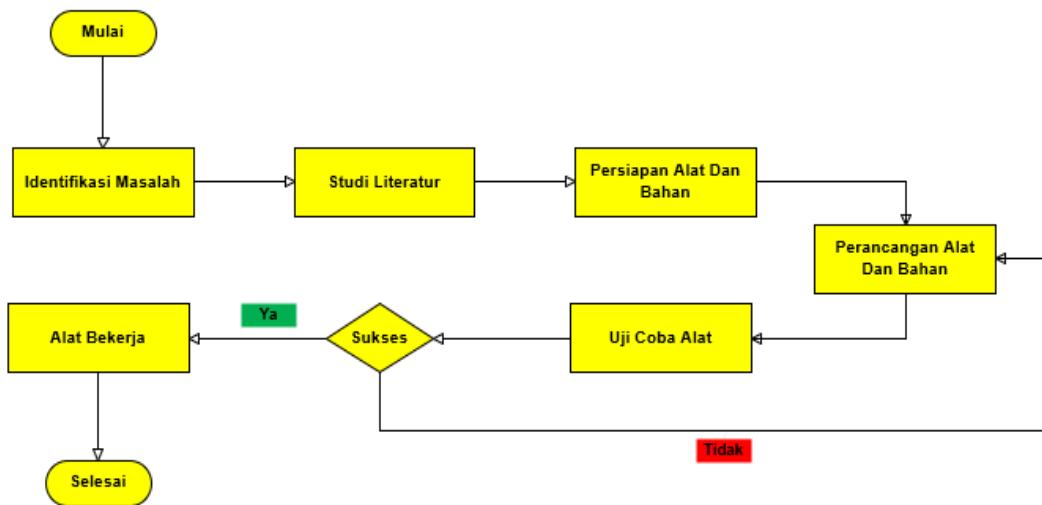


Gambar 1. Metode *Prototypye*

Metode prototype memiliki tiga tahapan sebagai berikut :

- Pada tahap pertama yaitu “Listen to Customer” yang merupakan proses komunikasi antara pengguna sepeda motor dengan pengembang yang dapat langsung diterapkan sesuai dengan keinginan dari pengguna sepeda motor.
- Selanjutnya masuk tahap “Build/Revise Mock-Up” yaitu pembuatan pemodelan setengah jadi.
- Pada tahap ketiga adalah “Customer Test Drives Mock-Up” yang merupakan suatu kegiatan pengujian program yang dilakukan oleh customer. Apabila terdapat keinginan pengguna sepeda motor yang belum tercapai atau ada bagian yang ingin ditambahkan dari sistem yang dikembangkan maka aktivitas kembali dilanjutkan ke tahap semula yaitu “Listen to Customer”.

Langkah-langkah dari metode di atas juga digambarkan peneliti dalam diagram alur berikut :



Gambar 2. Alur Penelitian

Dari gambar diatas diagram alur penelitian dapat dijelaskan sebagai berikut :

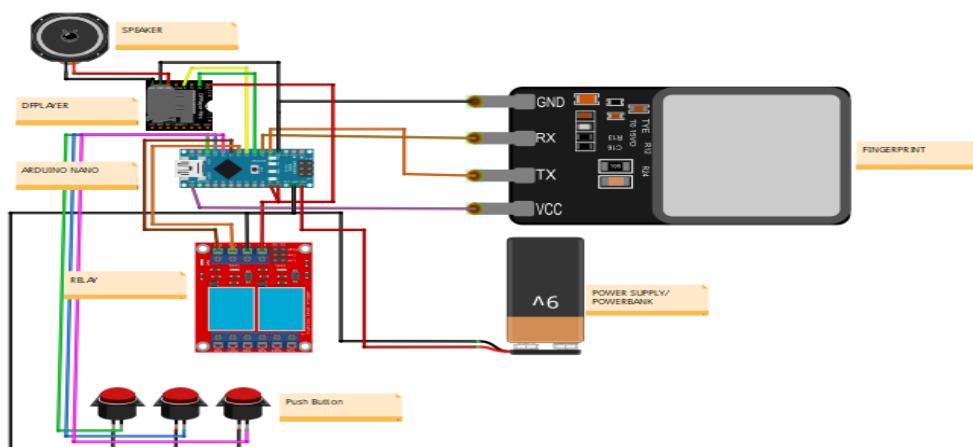
- Pada tahapan pertama adalah identifikasi masalah pada sepeda motor yang merupakan proses tahapan yang paling penting serta bertujuan untuk menguraikan dan menganalisa masalah yang nantinya akan menentukan penelitian.
- Pada tahapan kedua yaitu studi literatur, studi literatur merupakan cara memahami dan mempelajari teori yang masih relevan serta berhubungan yang terakit dengan masalah yang akan nantinya diselesaikan. Teori yang relevan digunakan untuk memecahkan masalah tersebut, kebanyakan berada di internet dan berbagai sumber lainnya seperti buku tentang sepeda motor dan keamanan berbasis fingerprint, jurnal yang terkait dengan rancang bangun sistem keamanan sepeda motor menggunakan fingerprint atau studi literatur yang dilakukan untuk mendapatkan pemahaman untuk pemecahan masalah yang pada nantinya akan diselesaikan.

- c. Pada tahap ketiga peneliti akan menyiapkan perangkat lunak dan perangkat keras yang di gunakan untuk merangkai alat. Adapun perangkat lunak dan perangkat keras yang digunakan dalam penelitian ini adalah :
1. Perangkat Lunak (Software)
    - a) Aplikasi Arduino IDE
    - b) Sistem Operasi Windows 10 64 Bit
    - c) Fritzing
  2. Perangkat Keras (Hardware)
    - a) Laptop
    - b) Arduino Nano
    - c) DFPlayer
    - d) Kartu Memori
    - e) Push Button
    - f) Speaker
    - g) Sensor Sidik Jari
    - h) Powerbank
    - i) PCB (Printed Circuit Board)
    - j) Relay 2 Channel
- d. Pada tahap ke empat perancangan dan pembuatan alat. Dalam perancangan dan pembuatan alat ini terdapat dua bagian perangkat yang dibutuhkan yaitu perangkat keras (Hardware) dan perangkat lunak (Software) :
1. Perangkat Keras (Hardware) yang akan digunakan berikut perancangan perangkat keras (Hardware) yang digunakan
    - a) Arduino Nano, sebagai mikrokontroler.
    - b) Sensor Fingerprint, sebagai pendekripsi sidik jari.
    - c) DFPlayer, sebagai alat untuk memutar suara/audio
    - d) Kartu memori digunakan untuk menyimpan file suara/audio
    - e) Speaker, sebagai alat untuk mengeluarkan suara/audio dari DFPlayer.
    - f) Relay 2 Channel, sebagai penghubung dan pemutus arus listrik yang ada pada kontak dan stater pada sepeda motor.
    - g) Powerbank, sebagai sumber arus listrik.
    - h) Push button digunakan sebagai tombol untuk emergency serta untuk menambah sidik jari dan menghapus sidik jari.
    - i) PCB, tempat untuk memasang Arduino Nano, DFPlayer, Relay.
    - j) Black box, sebagai tempat untuk menaruh komponen yang telah terpasang pada PCB.
  2. Perangkat lunak (Software) yang digunakan oleh peneliti dalam membuat program yang akan di implementasikan ke dalam alat. Ini dilakukan agar program yang telah dirancang dapat mengatur sistem kerja hardware sesuai dengan perancangan serta bahasa pemrograman yang digunakan ialah Arduino IDE dan Fritzing.
  - e. Tahap selanjutnya ialah pengujian alat. Selama tahap pengujian alat ini, operasi ini akan dilakukan untuk mengetahui apakah alat yang telah kerjakan beroperasi sesuai dengan rencana. Pengujian yang dilakukan adalah pengujian dengan menggunakan sidik jari, jika berhasil maka alat bekerja, apabila tidak memenuhi rencana awal maka akan kembali ke tahap perancangan dan pembuatan alat.

### 3. HASIL DAN PEMBAHASAN

#### 3.1. Analisis Sistem

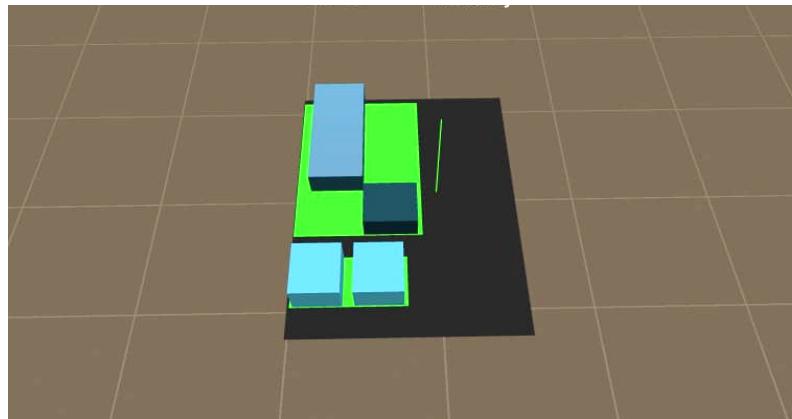
Analisis sistem dapat dilihat pada gambar 3



Gambar 3. Analisis Sistem

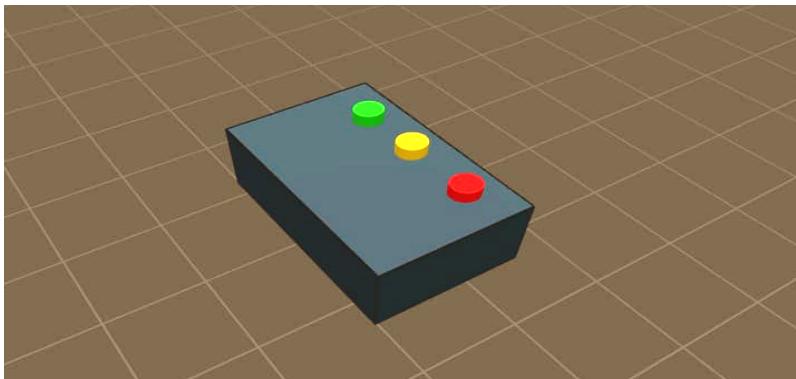
### 3.2. Tahap Perancangan

Pada tahap perancangan ini dilakukan terlebih dahulu dengan melakukan desain alat kemudian dilakukan desain coding sebagai berikut.



Gambar 4. Tampak dari dalam

Dapat dilihat di gambar 4 komponen A ialah arduino nano, komponen B ialah dfplayer, komponen C ialah relay 1, komponen D ialah relay 2 dan komponen E ialah PCB.



Gambar 5. Tampak dari luar

Pada gambar 5 ialah tampak luar dari alat dimana ada tiga push button yang berbeda warna. Dimana push button berwarna hijau berfungsi untuk menambahkan sidik jari, push button berwarna kuning berfungsi untuk menghapus sidik jari dan push button berwarna merah berfungsi sebagai tombol emergency.

The screenshot shows the Arduino IDE interface with the code for the fingerprint lock. The code is written in C++ and includes functions for handling serial communication and managing fingerprints. It uses variables like `Serial`, `fp`, `fpData`, and `fpIndex`. The code handles various commands from a fingerprint sensor and prints responses to the serial monitor.

```
File Edit Sketch Tools Help
Arduino Nano
Signatures
1:16
261: if (p == FINGERPRINT_PACKETRECIEVEERR) {
262:     Serial.println("Communication error");
263:     return p;
264: }
265:
266: else if (p == FINGERPRINT_NOFINGER) {
267:     Serial.println("No finger found");
268:     fpIndex = 0;
269:     fpData[0] = 0;
270: }
271: else if (p == FINGERPRINT_PACKETCOMPLETE) {
272:     Serial.println("Fingerprint received");
273:     fpIndex = 0;
274:     fpData[0] = 0;
275: }
276: else if (p == FINGERPRINT_FLASHING) {
277:     Serial.println("Fingerprint flash");
278:     fpIndex = 0;
279:     fpData[0] = 0;
280: }
281: else if (p == FINGERPRINT_ENROLLING) {
282:     Serial.println("Fingerprint enrollment");
283:     fpIndex = 0;
284:     fpData[0] = 0;
285: }
286: else if (p == FINGERPRINT_VERIFIED) {
287:     Serial.println("Fingerprint verified");
288:     fpIndex = 0;
289:     fpData[0] = 0;
290: }
291: else if (p == FINGERPRINT_STOREFAIL) {
292:     Serial.println("Fingerprint store fail");
293:     fpIndex = 0;
294:     fpData[0] = 0;
295: }
296: else if (p == FINGERPRINT_FLASHFAIL) {
297:     Serial.println("Fingerprint flash fail");
298:     fpIndex = 0;
299:     fpData[0] = 0;
300: }
301: else if (p == FINGERPRINT_ERROR_NOMATCH) {
302:     Serial.println("Fingerprint match fail");
303:     fpIndex = 0;
304:     fpData[0] = 0;
305: }
306: else if (p == FINGERPRINT_ERROR_INVALIDDATA) {
307:     Serial.println("Fingerprint invalid data");
308:     fpIndex = 0;
309:     fpData[0] = 0;
310: }
311: else if (p == FINGERPRINT_ERROR_FILEFORMAT) {
312:     Serial.println("Fingerprint file format error");
313:     fpIndex = 0;
314:     fpData[0] = 0;
315: }
316: else if (p == FINGERPRINT_ERROR_NOCARD) {
317:     Serial.println("Fingerprint no card");
318:     fpIndex = 0;
319:     fpData[0] = 0;
320: }
321: else if (p == FINGERPRINT_ERROR_TIMEOUT) {
322:     Serial.println("Fingerprint timeout");
323:     fpIndex = 0;
324:     fpData[0] = 0;
325: }
326: else if (p == FINGERPRINT_ERROR_AFINGER) {
327:     Serial.println("Fingerprint too many fingers");
328:     fpIndex = 0;
329:     fpData[0] = 0;
330: }
331: else if (p == FINGERPRINT_ERROR_NOFINGER) {
332:     Serial.println("Fingerprint no finger");
333:     fpIndex = 0;
334:     fpData[0] = 0;
335: }
336: else if (p == FINGERPRINT_ERROR_AFINGER2) {
337:     Serial.println("Fingerprint too many fingers");
338:     fpIndex = 0;
339:     fpData[0] = 0;
340: }
341: else if (p == FINGERPRINT_ERROR_NOCARD2) {
342:     Serial.println("Fingerprint no card");
343:     fpIndex = 0;
344:     fpData[0] = 0;
345: }
346: else if (p == FINGERPRINT_ERROR_AFINGER3) {
347:     Serial.println("Fingerprint too many fingers");
348:     fpIndex = 0;
349:     fpData[0] = 0;
350: }
351: else if (p == FINGERPRINT_ERROR_NOCARD3) {
352:     Serial.println("Fingerprint no card");
353:     fpIndex = 0;
354:     fpData[0] = 0;
355: }
356: else if (p == FINGERPRINT_ERROR_AFINGER4) {
357:     Serial.println("Fingerprint too many fingers");
358:     fpIndex = 0;
359:     fpData[0] = 0;
360: }
361: else if (p == FINGERPRINT_ERROR_NOCARD4) {
362:     Serial.println("Fingerprint no card");
363:     fpIndex = 0;
364:     fpData[0] = 0;
365: }
366: else if (p == FINGERPRINT_ERROR_AFINGER5) {
367:     Serial.println("Fingerprint too many fingers");
368:     fpIndex = 0;
369:     fpData[0] = 0;
370: }
371: else if (p == FINGERPRINT_ERROR_NOCARD5) {
372:     Serial.println("Fingerprint no card");
373:     fpIndex = 0;
374:     fpData[0] = 0;
375: }
376: else if (p == FINGERPRINT_ERROR_AFINGER6) {
377:     Serial.println("Fingerprint too many fingers");
378:     fpIndex = 0;
379:     fpData[0] = 0;
380: }
381: else if (p == FINGERPRINT_ERROR_NOCARD6) {
382:     Serial.println("Fingerprint no card");
383:     fpIndex = 0;
384:     fpData[0] = 0;
385: }
386: else if (p == FINGERPRINT_ERROR_AFINGER7) {
387:     Serial.println("Fingerprint too many fingers");
388:     fpIndex = 0;
389:     fpData[0] = 0;
390: }
391: else if (p == FINGERPRINT_ERROR_NOCARD7) {
392:     Serial.println("Fingerprint no card");
393:     fpIndex = 0;
394:     fpData[0] = 0;
395: }
396: else if (p == FINGERPRINT_ERROR_AFINGER8) {
397:     Serial.println("Fingerprint too many fingers");
398:     fpIndex = 0;
399:     fpData[0] = 0;
400: }
401: else if (p == FINGERPRINT_ERROR_NOCARD8) {
402:     Serial.println("Fingerprint no card");
403:     fpIndex = 0;
404:     fpData[0] = 0;
405: }
406: else if (p == FINGERPRINT_ERROR_AFINGER9) {
407:     Serial.println("Fingerprint too many fingers");
408:     fpIndex = 0;
409:     fpData[0] = 0;
410: }
411: else if (p == FINGERPRINT_ERROR_NOCARD9) {
412:     Serial.println("Fingerprint no card");
413:     fpIndex = 0;
414:     fpData[0] = 0;
415: }
416: else if (p == FINGERPRINT_ERROR_AFINGER10) {
417:     Serial.println("Fingerprint too many fingers");
418:     fpIndex = 0;
419:     fpData[0] = 0;
420: }
421: else if (p == FINGERPRINT_ERROR_NOCARD10) {
422:     Serial.println("Fingerprint no card");
423:     fpIndex = 0;
424:     fpData[0] = 0;
425: }
426: else if (p == FINGERPRINT_ERROR_AFINGER11) {
427:     Serial.println("Fingerprint too many fingers");
428:     fpIndex = 0;
429:     fpData[0] = 0;
430: }
431: else if (p == FINGERPRINT_ERROR_NOCARD11) {
432:     Serial.println("Fingerprint no card");
433:     fpIndex = 0;
434:     fpData[0] = 0;
435: }
436: else if (p == FINGERPRINT_ERROR_AFINGER12) {
437:     Serial.println("Fingerprint too many fingers");
438:     fpIndex = 0;
439:     fpData[0] = 0;
440: }
441: else if (p == FINGERPRINT_ERROR_NOCARD12) {
442:     Serial.println("Fingerprint no card");
443:     fpIndex = 0;
444:     fpData[0] = 0;
445: }
446: else if (p == FINGERPRINT_ERROR_AFINGER13) {
447:     Serial.println("Fingerprint too many fingers");
448:     fpIndex = 0;
449:     fpData[0] = 0;
450: }
451: else if (p == FINGERPRINT_ERROR_NOCARD13) {
452:     Serial.println("Fingerprint no card");
453:     fpIndex = 0;
454:     fpData[0] = 0;
455: }
456: else if (p == FINGERPRINT_ERROR_AFINGER14) {
457:     Serial.println("Fingerprint too many fingers");
458:     fpIndex = 0;
459:     fpData[0] = 0;
460: }
461: else if (p == FINGERPRINT_ERROR_NOCARD14) {
462:     Serial.println("Fingerprint no card");
463:     fpIndex = 0;
464:     fpData[0] = 0;
465: }
466: else if (p == FINGERPRINT_ERROR_AFINGER15) {
467:     Serial.println("Fingerprint too many fingers");
468:     fpIndex = 0;
469:     fpData[0] = 0;
470: }
471: else if (p == FINGERPRINT_ERROR_NOCARD15) {
472:     Serial.println("Fingerprint no card");
473:     fpIndex = 0;
474:     fpData[0] = 0;
475: }
476: else if (p == FINGERPRINT_ERROR_AFINGER16) {
477:     Serial.println("Fingerprint too many fingers");
478:     fpIndex = 0;
479:     fpData[0] = 0;
480: }
481: else if (p == FINGERPRINT_ERROR_NOCARD16) {
482:     Serial.println("Fingerprint no card");
483:     fpIndex = 0;
484:     fpData[0] = 0;
485: }
486: else if (p == FINGERPRINT_ERROR_AFINGER17) {
487:     Serial.println("Fingerprint too many fingers");
488:     fpIndex = 0;
489:     fpData[0] = 0;
490: }
491: else if (p == FINGERPRINT_ERROR_NOCARD17) {
492:     Serial.println("Fingerprint no card");
493:     fpIndex = 0;
494:     fpData[0] = 0;
495: }
496: else if (p == FINGERPRINT_ERROR_AFINGER18) {
497:     Serial.println("Fingerprint too many fingers");
498:     fpIndex = 0;
499:     fpData[0] = 0;
500: }
501: else if (p == FINGERPRINT_ERROR_NOCARD18) {
502:     Serial.println("Fingerprint no card");
503:     fpIndex = 0;
504:     fpData[0] = 0;
505: }
506: else if (p == FINGERPRINT_ERROR_AFINGER19) {
507:     Serial.println("Fingerprint too many fingers");
508:     fpIndex = 0;
509:     fpData[0] = 0;
510: }
511: else if (p == FINGERPRINT_ERROR_NOCARD19) {
512:     Serial.println("Fingerprint no card");
513:     fpIndex = 0;
514:     fpData[0] = 0;
515: }
516: else if (p == FINGERPRINT_ERROR_AFINGER20) {
517:     Serial.println("Fingerprint too many fingers");
518:     fpIndex = 0;
519:     fpData[0] = 0;
520: }
521: else if (p == FINGERPRINT_ERROR_NOCARD20) {
522:     Serial.println("Fingerprint no card");
523:     fpIndex = 0;
524:     fpData[0] = 0;
525: }
526: else if (p == FINGERPRINT_ERROR_AFINGER21) {
527:     Serial.println("Fingerprint too many fingers");
528:     fpIndex = 0;
529:     fpData[0] = 0;
530: }
531: else if (p == FINGERPRINT_ERROR_NOCARD21) {
532:     Serial.println("Fingerprint no card");
533:     fpIndex = 0;
534:     fpData[0] = 0;
535: }
536: else if (p == FINGERPRINT_ERROR_AFINGER22) {
537:     Serial.println("Fingerprint too many fingers");
538:     fpIndex = 0;
539:     fpData[0] = 0;
540: }
541: else if (p == FINGERPRINT_ERROR_NOCARD22) {
542:     Serial.println("Fingerprint no card");
543:     fpIndex = 0;
544:     fpData[0] = 0;
545: }
546: else if (p == FINGERPRINT_ERROR_AFINGER23) {
547:     Serial.println("Fingerprint too many fingers");
548:     fpIndex = 0;
549:     fpData[0] = 0;
550: }
551: else if (p == FINGERPRINT_ERROR_NOCARD23) {
552:     Serial.println("Fingerprint no card");
553:     fpIndex = 0;
554:     fpData[0] = 0;
555: }
556: else if (p == FINGERPRINT_ERROR_AFINGER24) {
557:     Serial.println("Fingerprint too many fingers");
558:     fpIndex = 0;
559:     fpData[0] = 0;
560: }
561: else if (p == FINGERPRINT_ERROR_NOCARD24) {
562:     Serial.println("Fingerprint no card");
563:     fpIndex = 0;
564:     fpData[0] = 0;
565: }
566: else if (p == FINGERPRINT_ERROR_AFINGER25) {
567:     Serial.println("Fingerprint too many fingers");
568:     fpIndex = 0;
569:     fpData[0] = 0;
570: }
571: else if (p == FINGERPRINT_ERROR_NOCARD25) {
572:     Serial.println("Fingerprint no card");
573:     fpIndex = 0;
574:     fpData[0] = 0;
575: }
576: else if (p == FINGERPRINT_ERROR_AFINGER26) {
577:     Serial.println("Fingerprint too many fingers");
578:     fpIndex = 0;
579:     fpData[0] = 0;
580: }
581: else if (p == FINGERPRINT_ERROR_NOCARD26) {
582:     Serial.println("Fingerprint no card");
583:     fpIndex = 0;
584:     fpData[0] = 0;
585: }
586: else if (p == FINGERPRINT_ERROR_AFINGER27) {
587:     Serial.println("Fingerprint too many fingers");
588:     fpIndex = 0;
589:     fpData[0] = 0;
590: }
591: else if (p == FINGERPRINT_ERROR_NOCARD27) {
592:     Serial.println("Fingerprint no card");
593:     fpIndex = 0;
594:     fpData[0] = 0;
595: }
596: else if (p == FINGERPRINT_ERROR_AFINGER28) {
597:     Serial.println("Fingerprint too many fingers");
598:     fpIndex = 0;
599:     fpData[0] = 0;
600: }
601: else if (p == FINGERPRINT_ERROR_NOCARD28) {
602:     Serial.println("Fingerprint no card");
603:     fpIndex = 0;
604:     fpData[0] = 0;
605: }
606: else if (p == FINGERPRINT_ERROR_AFINGER29) {
607:     Serial.println("Fingerprint too many fingers");
608:     fpIndex = 0;
609:     fpData[0] = 0;
610: }
611: else if (p == FINGERPRINT_ERROR_NOCARD29) {
612:     Serial.println("Fingerprint no card");
613:     fpIndex = 0;
614:     fpData[0] = 0;
615: }
616: else if (p == FINGERPRINT_ERROR_AFINGER30) {
617:     Serial.println("Fingerprint too many fingers");
618:     fpIndex = 0;
619:     fpData[0] = 0;
620: }
621: else if (p == FINGERPRINT_ERROR_NOCARD30) {
622:     Serial.println("Fingerprint no card");
623:     fpIndex = 0;
624:     fpData[0] = 0;
625: }
626: else if (p == FINGERPRINT_ERROR_AFINGER31) {
627:     Serial.println("Fingerprint too many fingers");
628:     fpIndex = 0;
629:     fpData[0] = 0;
630: }
631: else if (p == FINGERPRINT_ERROR_NOCARD31) {
632:     Serial.println("Fingerprint no card");
633:     fpIndex = 0;
634:     fpData[0] = 0;
635: }
636: else if (p == FINGERPRINT_ERROR_AFINGER32) {
637:     Serial.println("Fingerprint too many fingers");
638:     fpIndex = 0;
639:     fpData[0] = 0;
640: }
641: else if (p == FINGERPRINT_ERROR_NOCARD32) {
642:     Serial.println("Fingerprint no card");
643:     fpIndex = 0;
644:     fpData[0] = 0;
645: }
646: else if (p == FINGERPRINT_ERROR_AFINGER33) {
647:     Serial.println("Fingerprint too many fingers");
648:     fpIndex = 0;
649:     fpData[0] = 0;
650: }
651: else if (p == FINGERPRINT_ERROR_NOCARD33) {
652:     Serial.println("Fingerprint no card");
653:     fpIndex = 0;
654:     fpData[0] = 0;
655: }
656: else if (p == FINGERPRINT_ERROR_AFINGER34) {
657:     Serial.println("Fingerprint too many fingers");
658:     fpIndex = 0;
659:     fpData[0] = 0;
660: }
661: else if (p == FINGERPRINT_ERROR_NOCARD34) {
662:     Serial.println("Fingerprint no card");
663:     fpIndex = 0;
664:     fpData[0] = 0;
665: }
666: else if (p == FINGERPRINT_ERROR_AFINGER35) {
667:     Serial.println("Fingerprint too many fingers");
668:     fpIndex = 0;
669:     fpData[0] = 0;
670: }
671: else if (p == FINGERPRINT_ERROR_NOCARD35) {
672:     Serial.println("Fingerprint no card");
673:     fpIndex = 0;
674:     fpData[0] = 0;
675: }
676: else if (p == FINGERPRINT_ERROR_AFINGER36) {
677:     Serial.println("Fingerprint too many fingers");
678:     fpIndex = 0;
679:     fpData[0] = 0;
680: }
681: else if (p == FINGERPRINT_ERROR_NOCARD36) {
682:     Serial.println("Fingerprint no card");
683:     fpIndex = 0;
684:     fpData[0] = 0;
685: }
686: else if (p == FINGERPRINT_ERROR_AFINGER37) {
687:     Serial.println("Fingerprint too many fingers");
688:     fpIndex = 0;
689:     fpData[0] = 0;
690: }
691: else if (p == FINGERPRINT_ERROR_NOCARD37) {
692:     Serial.println("Fingerprint no card");
693:     fpIndex = 0;
694:     fpData[0] = 0;
695: }
696: else if (p == FINGERPRINT_ERROR_AFINGER38) {
697:     Serial.println("Fingerprint too many fingers");
698:     fpIndex = 0;
699:     fpData[0] = 0;
700: }
690 Done uploading
```

Gambar 6. Koding alat

Adapun Programnya dapat dilihat pada Gambar 6, program di atas merupakan inti dari sebuah alat yang dimana fungsi program ini mengontrol sistem kerja alat yang dibuat, dengan adanya program ini alat yang dibuat dapat berjalan sesuai dengan rancangan dari alat

### 3.3. Pengujian

Pengujian ini dilaksanakan untuk menilai waktu respons ketika mengaktifkan motor menggunakan sensor sidik jari, dengan melakukan uji coba sebanyak 20 kali.

Tabel 1. Respon Sidik Jari

No	Status Sidik Jari	Respon (Detik)	Keterangan
1	Terdaftar	17,79	Menyala
2	Terdaftar	15,67	Menyala
3	Terdaftar	19,65	Menyala
4	Terdaftar	14,77	Menyala
5	Terdaftar	15,88	Menyala
6	Terdaftar	17,55	Menyala
7	Terdaftar	14,31	Menyala
8	Terdaftar	16,30	Menyala
9	Terdaftar	15,32	Menyala
10	Terdaftar	14,40	Menyala
11	Tidak Terdaftar	02,70	Tidak
12	Tidak Terdaftar	02,46	Tidak
13	Tidak Terdaftar	02,69	Tidak
14	Tidak Terdaftar	02,75	Tidak
15	Tidak Terdaftar	02,79	Tidak
16	Tidak Terdaftar	03,03	Tidak
17	Tidak Terdaftar	03,31	Tidak
18	Tidak Terdaftar	02,41	Tidak
19	Tidak Terdaftar	03,09	Tidak
20	Tidak Terdaftar	02,54	Tidak

Pengujian sidik jari sebagai inisiator untuk menghidupkan mesin motor dilaksanakan sebanyak 20 kali, terbagi menjadi 10 kali untuk sidik jari terdaftar dan 10 kali untuk sidik jari yang tidak terdaftar. Tabel 1 menampilkan hasil dari pengujian ini, yang bertujuan untuk memverifikasi kelancaran operasi sistem. Hasil dari 20 uji coba menunjukkan kondisi yang konsisten dengan program. Pada pengujian ini, delay rata-rata dari status fingerprint untuk sidik jari terdaftar adalah 16,16 detik, sementara delay rata-rata dari status fingerprint untuk sidik jari tidak terdaftar adalah 2,77 detik. Selain itu, selama pengujian ini, terbukti bahwa sistem mampu membaca sidik jari dengan baik dalam 20 percobaan, baik pada kondisi sidik jari terdaftar maupun tidak terdaftar. Dengan demikian, dapat disimpulkan bahwa keakuratan sensor fingerprint mencapai tingkat akurasi sebesar 100%.

## 4. KESIMPULAN

1. Rancangan yang dibuat telah berhasil di lakukan dengan hasil pengujian sesuai dengan harapan peneliti.
2. Sensor sidik jari akan berfungsi optimal apabila sidik jari yang ditempelkan dalam keadaan kering.
3. Sensor sidik jari dapat menunjukkan respons yang lambat atau bahkan tidak responsif sama sekali jika sidik jari dalam keadaan basah, berminyak, berdebu, kotor, atau terluka.
4. Meskipun telah diuji beberapa kali dengan berbagai pengguna, perangkat ini tetap beroperasi dengan baik.
5. Waktu respons pembacaan saat menempelkan sidik jari di sensor memerlukan waktu 1-2 detik.

## DAFTAR PUSTAKA

- [1] A. Surahman, A. T. Prastowo, and L. A. Aziz, "Rancang Alat Keamanan Sepeda Motor Honda Beat Berbasis Sim Gsm Menggunakan Metode Rancang Bangun," *J. Teknol. dan Sist. Tertanam*, vol. 3, no. 1, 2022, doi: 10.33365/jtst.v3i1.1918.
- [2] M. N. Ikhsan, I. Ibrahim, and R. Rahmadewi, "Sistem Keamanan Sepeda Motor dengan Teknologi Biometrik Sidik Jari Menggunakan Sensor Fingerprint R305," *STRING (Satuan Tulisan Ris. dan Inov. Teknol.)*, vol. 7, no. 2, p. 144, 2022, doi: 10.30998/string.v7i2.13797.
- [3] B. P. Statistik, "Statistik Kriminal Statistik Kriminal," 2022.
- [4] F. A. Wenando *et al.*, "Aplikasi dan Kerentanan Algoritma Probabilistic Neural Network ( PNN ) : Systematic Literature Systematic Literature Review : Applications and Vulnerabilities of Probabilistic Neural Network," *J. CoSciTech (Computer Sci. Inf. Technol.)*, vol. 4, no. 2, pp. 491–499, 2023.
- [5] M. F. Ramadhan and C. Bella, "Penggunaan Sensor Sidik Jari Untuk Keamanan Sepeda Motor Menggunakan Arduino," *J. Portal Data*, vol. 1, no. 3, pp. 1–19, 2021, [Online]. Available: <http://portaldatal.org/index.php/portaldatal/article/view/58%0Ahttp://portaldatal.org/index.php/portaldatal/article/download/58/58>
- [6] R. Ahmad and L. Kerta Wijaya, "Sistem Kontrol Kendaraan Roda Dua Menggunakan Mikrokontroler Berbasis Fingerprint," *Infotek J. Inform. dan Teknol.*, vol. 2, no. 2, pp. 127–134, 2019.
- [7] F. Wanto, A., Sulaiman, O. K., Ginantra, N. L. W. S. R., Simarmata, J., Lubis, M. R., & Effendy, "Biometrika: Teknologi Identifikasi," *Yayasan Kita Menulis*, 2020.

- [8] Muhamad Satibi Mulya, I. Yustiana, and I. Lucia Khrisma, “Rancang Bangun Sistem Keamanan dan Monitoring Kendaraan Berbasis IoT dan Mobile Apps,” *J. CoSciTech (Computer Sci. Inf. Technol.)*, vol. 3, no. 2, pp. 58–65, 2022, doi: 10.37859/coscitech.v3i2.3934.
- [9] A. Syarifudin, “Perancangan Sistem Informasi Pengajuan dan Pelaporan Pembayaran Tunjangan Kinerja Kementerian Keuangan Menggunakan Metode Prototype,” *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 8, no. 2, pp. 149–158, 2019, doi: 10.32736/sisfokom.v8i2.641.
- [10] S. Sujono, M. S. Mayasari, and K. Koloniawan, “Prototipe Aplikasi Simpan Pinjam Pada Koperasi Darma Karya Pangkalpinang Babel,” *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 8, no. 1, pp. 68–73, 2019, doi: 10.32736/sisfokom.v8i1.609.