

BAB II KAJIAN PUSTAKA

2.1. Penelitian sebelumnya

Beberapa penelitian telah dilakukan berkaitan dengan penggunaan sensor *fingerprint* dalam sistem absensi dan penggunaan Raspberry Pi sebagai prosesor. Diantara penelitian yang dilakukan adalah penelitian oleh Arun Raja dan Arunkumar tentang pengidentifikasian sidik jari menggunakan aplikasi GUI berbasis Linux. Selain itu, Jordi Sapes dan Francesc Solsona melakukan penelitian tentang penggunaan Raspberry Pi dalam memvalidasi pengguna dengan memindai sidik jari untuk sistem *security*.

Arun Raja dan Arunkumar melakukan penelitian yang berjudul “*Biometrics Authentication Using Raspberry Pi*”^[2]. Dalam penelitian ini sensor *fingerprint* yang digunakan adalah U.are.U 4500. Penelitian ini berfokus pada proses pengidentifikasian sidik jari yang terdeteksi oleh sensor *fingerprint* menggunakan aplikasi Fprint Demo dan *image processing*. Fprint Demo adalah aplikasi untuk mendaftarkan sidik jari yang kemudian disimpan di *database*. *Database* yang digunakan pada penelitian ini adalah Postgresql. Sedangkan *image processing* digunakan untuk pencocokan sidik jari yang tersimpan di *database* dengan sidik jari yang terbaca oleh sensor.

Dalam pendeteksian sidik jari menggunakan sensor *fingerprint* dan *image processing*, Arun Raja dan Arunkumar menggunakan algoritma sebagai berikut.

Step 1: hubungkan sensor *fingerprint* dengan Raspberry Pi

Step 2: buka Fprint Demo

Step 3: baca dan simpan *image template*

Step 4: konversi *image template* ke *binary template*

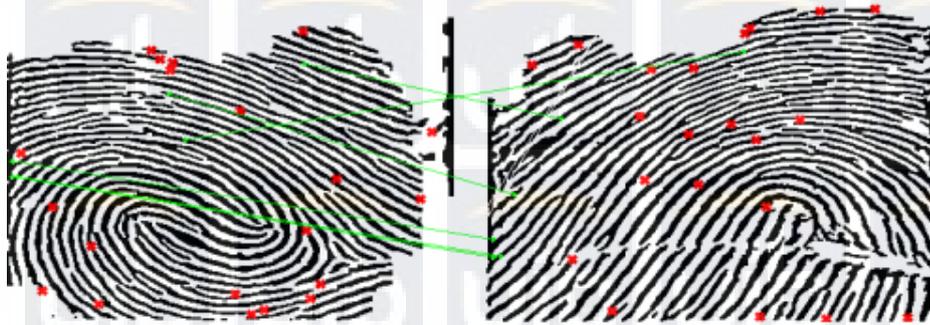
Step 5: simpan *template* yang didapatkan di *database* Postgresql

Step 6: perbandingan antara berbagai *template* dilakukan dengan Python

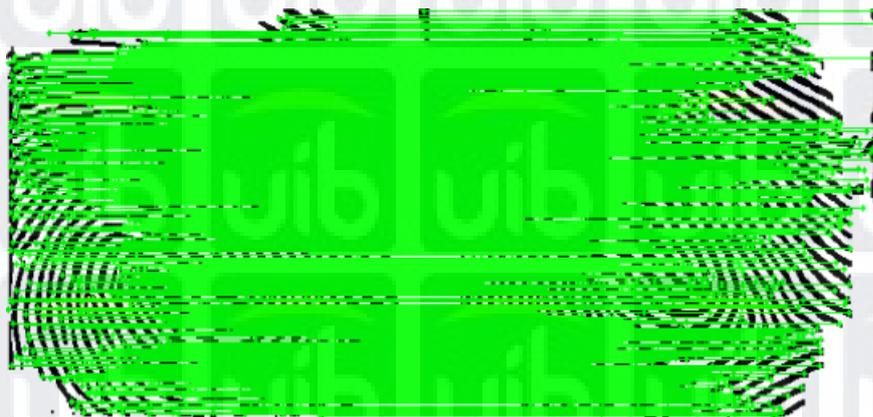
Step 7: menentukan titik kecocokan antara gambar terdeteksi dengan *template* menggunakan algoritma *image processing* FLANN_INDEX_KDTREE.

Hasil dari pendeteksian sidik jari menggunakan Fprint Demo dengan *image processing* adalah titik-titik kecocokan antara gambar sidik jari yang terdeteksi dengan *template* sidik jari yang tersimpan di *database*. Setiap titik kecocokan yang

ditemukan antara sidik jari yang disimpan di *database* dengan sidik jari yang terdeteksi sensor ditandai dengan garis berwarna hijau. Gambar pencocokan sidik jari dapat dilihat pada Gambar 2.1 untuk sidik jari yang tidak cocok dan Gambar 2.2 untuk sidik jari yang cocok.



Gambar 2.1. Sidik Jari tidak cocok
(Sumber: Arun Raja dan Arunkumar. 2015)



Gambar 2.2. Sidik Jari cocok
(Sumber: Arun Raja dan Arunkumar. 2015)

Pada Gambar 2.1 hanya sedikit kecocokan yang ditemukan antara gambar yang terdeteksi sensor dan *template* sidik jari yang tersimpan di *database*. Dengan demikian diputuskan bahwa sidik jari tidak cocok dengan *template*. Sedangkan pada Gambar 2.2 banyak kecocokan yang ditemukan antara sidik jari yang terdeteksi dengan sidik jari yang tersimpan di *database*. Dengan demikian diputuskan bahwa sidik jari cocok dengan *template*. Dari penelitian Arun Raja dan Arunkumar dapat disimpulkan bahwa rancangan sistem yang mereka ajukan mampu mengidentifikasi perbedaan antara sidik jari yang cocok dengan sidik jari yang tidak cocok.

Jordi Sapes dan Francesc Solsona melakukan penelitian untuk penggunaan sensor *fingerprint* GT511C1R dan Raspberry Pi dalam penelitian yang berjudul “FingerScanner: Embedding a Fingerprint Scanner in a Raspberry Pi”^[3]. Penelitian ini dilakukan untuk pengembangan sistem *security* dengan memvalidasi pengguna yang terdaftar melalui identifikasi sidik jari. Sistem *security* ini dibuat dengan model pola *client-server* dimana Raspberry Pi difungsikan sebagai *server* dan satu unit PC bertindak sebagai *client*. Diantara pengaplikasian penelitian ini yaitu pada kunci magnetik pintu yang terhubung ke sistem *security* sehingga yang bisa mengakses pintu adalah orang yang terdaftar.

Pada penelitian ini, server menyimpan data sidik jari yang terdaftar pada *database* SQLite. Pada SQLite terdapat beberapa *method* yang digunakan diantaranya “GET” untuk mendapatkan data sidik jari, “POST” untuk menyimpan sidik jari baru, “PUT” untuk memperbarui data sidik jari, dan “DELETE” untuk menghapus data sidik jari. Proses utama pada komunikasi dengan SQLite yaitu perekaman sidik jari, penghapusan data sidik jari, pengidentifikasian sidik jari, dan pembaruan data sidik jari. Sidik jari yang tersimpan pada *database* ini digunakan sebagai pembanding dengan sidik jari yang terdeteksi oleh sensor. Apabila sidik jari yang terdeteksi cocok dengan *template* di *database* maka output GPIO akan membuka kunci magnetik selama 3 detik.

Pengujian dan analisa penelitian ini berfokus pada pengukuran kemampuan sistem komunikasi untuk mengetahui *response time* dan tingkah laku komponen-komponen yang berbeda pada sistem yaitu PC sebagai client, Raspberry pi sebagai server, dan sensor *fingerprint*. Hasil dari pengujian dibagi menjadi tiga kelompok yaitu *response time* akses SQLite, *response time* SQLite dan komunikasi dengan *scanner*, dan *response time* dengan pengaruh interaksi dengan *user*.

- Akses SQLite

Response time dari SQLite sangat cepat yaitu 0.113 detik dan 0.104 detik untuk mendapatkan informasi sidik jari dari SQLite. Proses pengidentifikasian sidik jari sendiri memerlukan waktu 4.5 detik

- Akses SQLite + komunikasi *scanner*

Komunikasi antara SQLite dengan *scanner* memerlukan waktu 2.413 detik untuk proses membuka *port* serial dan pengiriman serta penerimaan data. Waktu ini termasuk lama disebabkan oleh keterbatasan *Serial Communication Interface*.

- Akses SQLite + komunikasi *scanner* + *Human interactivity*

Response time ini menjadi yang paling lambat karena keterlibatan interaksi dengan *user*. Proses *enroll* memerlukan waktu 7.1 detik

Jordi dan Fransces menyimpulkan bahwa *Finger Scanner* yang dibuat dengan sensor *fingerprint* GT511C1R dan Raspberry Pi dapat berfungsi dan memiliki kecepatan yang baik setelah melakukan beberapa pengujian. Diantara pengujian yang dilakukan adalah kecepatan pendaftaran sidik jari dan pengidentifikasian sidik jari oleh sensor *fingerprint* GT511C1R masing-masing dengan rata-rata 7 detik dan 4 detik. Selain itu tingkat kemampuan sensor GT511C1R juga diuji dengan menghasilkan persentase keberhasilan pada rentang 70% - 80%.

Seperti yang dijelaskan pada latar belakang dari penelitian ini, Hasan Tok et al. mengembangkan sistem absensi *mobile* pada penelitian yang berjudul “A Novel ZigBee Based Mobile Fingerprint Student Attendance System”^[1]. Sistem absensi *mobile* tersebut dibuat dengan menggunakan Sensor *fingerprint*, Arduino, ZigBee, dan Raspberry Pi. Cara kerja sistem absensi *mobile* ini adalah sebagai berikut.

- Sensor *fingerprint* digunakan untuk mendeteksi sidik jari peserta absensi sedangkan Arduino digunakan untuk memproses data sensor untuk proses pengidentifikasian sidik jari.
- Setelah proses pengidentifikasian selesai, Arduino akan mengirim data ke Raspberry Pi melalui komunikasi serial ZigBee.
- Untuk melakukan komunikasi ini diperlukan transmitter ZigBee di sisi Arduino dan receiver ZigBee pada sisi Raspberry Pi.
- Data yang diterima Raspberry akan di input ke *database* web server untuk meng-*update* isi absensi.

Sistem absensi *mobile* serupa juga dikembangkan oleh Sadat Duraki et al. dalam penelitian berjudul “A Mobile Application for Wireless Attendance System”^[4]. Sama halnya dengan penelitian yang dilakukan oleh Hasan Tok et al., Sadat Duraki et al. juga menggunakan sistem yang dibuat dengan Sensor

fingerprint, Arduino, ZigBee, dan Raspberry Pi. Walaupun menggunakan hardware yang serupa, namun penelitian Sadat Duraki et al. menitik beratkan pada perancangan *database* web server yang mudah digunakan dengan tampilan pada *mode admin* dan *mode teacher*.

Dari kedua penelitian Hasan Tok et al. dan Sadat Duraki et al., penelitian sistem absensi *mobile* ini terdapat dua kali pengiriman data yaitu pengiriman data dari Arduino ke Raspberry Pi dan dari Raspberry Pi ke Web Server. Pengolahan data absensi sepenuhnya dilakukan pada web server melalui Raspberry Pi yang terhubung ke jaringan internet.

2.2. Sidik Jari

Sidik jari adalah biometrik yang telah teruji secara ekstensif dan tidak pernah berubah serta relatif mudah untuk diambil^[5]. Pola sidik jari bersifat permanen sehingga dari bayi sampai dewasa pola sidik jari tidak akan berubah. Sidik jari bersifat unik yaitu tidak ada dua sidik jari yang identik sama. Setiap orang memiliki sepuluh sidik jari tangan yang melekat pada dirinya dimana setiap jari memiliki sidik jari yang berbeda. Dengan demikian sidik jari digunakan secara luas untuk keperluan identifikasi.

Secara garis besar pola sidik jari dikelompokkan menjadi 4 pola dasar yaitu Whorl, Arch, Left Slant Loop, dan Right Slant Loop^[5]. Pengelompokan ini dibagi berdasarkan pola *ridge* pada sidik jari. *Ridge* adalah garis-garis halus pada sidik jari yang membentuk pola. Berikut ini adalah penjelasan pembagian pola pada sidik jari.

- Whorl adalah pola sidik jari yang menyerupai spiral. Pola sidik jari Whorl ditemukan pada sekitar 30% dari seluruh sidik jari. Pola Whorl dapat dilihat pada Gambar 2.3.



Gambar 2.3. Pola sidik jari Whorl
(Sumber: DigitalPersona White Paper Guide to Fingerprint Recognition)

- Arch adalah pola sidik jari yang berbentuk seperti kurva. Pola Arch dapat dilihat pada Gambar 2.4.



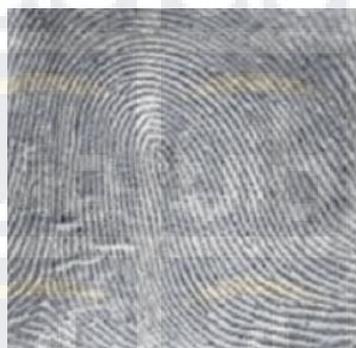
Gambar 2.4. Pola sidik jari Arch
(Sumber: DigitalPersona White Paper Guide to Fingerprint Recognition)

- Left Slant Loop adalah pola sidik jari berbentuk putaran yang condong ke kiri. Pola Left Slant Loop dapat dilihat pada Gambar 2.5.



Gambar 2.5. Pola sidik jari Left Slant Loop
(Sumber: Peter Komarinski. 2005)

- Right Slant Loop adalah pola sidik jari berbentuk putaran yang condong ke kanan. Pola Right Slant Loop dapat dilihat pada Gambar 2.6.



Gambar 2.6. Pola sidik jari Right Slant Loop
(Sumber: Peter Komarinski. 2005)

2.3. *Sensor Fingerprint*

Sensor *fingerprint* adalah alat deteksi sidik jari yang dibuat dengan *Digital Signal Processing* (DSP) untuk mengolah gambar, kalkulasi, dan kemampuan pencarian^[6]. Sensor *fingerprint* dihubungkan dengan dengan mikrokontroler yang memberikan perintah atau signal ke sensor untuk mengambil gambar sidik jari, mendeteksi sidik jari, atau mencocokkan sidik jari. Pada sensor *fingerprint* terdapat memori flash yang berguna untuk menyimpan data sidik jari yang didaftarkan. Jumlah sidik jari yang bisa disimpan tergantung seberapa besar kapasitas memori flash yang dipakai di sensor. Gambar sensor *fingerprint* dapat dilihat pada Gambar 2.7.



Gambar 2.7. Sensor *Fingerprint*.

(Sumber: <https://www.tokopedia.com/hinarichan/fpm10a-optical-fingerprint-scanner-sensor-module-arduino-locks?src=topads>)

Sensor *fingerprint* yang digunakan ada jenis sensor FPM10A. Sensor *fingerprint* ini memiliki beberapa fitur diantaranya adalah *Enrolling*, *Searching*, dan *Deleting*. Fitur-fitur ini diakses melalui mikrokontroler yang terprogram atau *software* Windows. *Enrolling* adalah fitur untuk mendaftarkan sidik jari atau menyimpan sidik jari pada ID tertentu sehingga bisa diakses dengan fitur *searching*. *Searching* adalah fitur untuk mencari ID sidik jari yang telah terdeteksi oleh sensor dan memberitahu ID sidik jari. *Deleting* adalah proses penghapusan data sidik jari dari memori flash sensor. Pada proses penghapusan data harus ditentukan ID sidik jari yang akan dihapus.

2.4. Raspberry Pi

Raspberry Pi adalah sebuah *Single Board Computer* (SBC) yang berarti komputer yang dibangun pada satu *Printed Circuit Board* (PCB). Ide awal dibalik munculnya Raspberry Pi adalah kekhawatiran para peneliti komputer dari

Universitas Cambridge karena menurunnya minat dan jumlah siswa yang tertarik untuk belajar ilmu komputer. Raspberry Pi dibuat dan dirancang oleh Raspberry Pi Foundation dari Inggris dengan maksud mendorong pembelajaran ilmu dasar komputer untuk pelajar sekolah dan setiap orang yang tertarik pada perangkat keras komputer, pemrograman, dan proyek DIY (*Do-it Yourself*)^[7].

Raspberry Pi memungkinkan berbagai orang dari kalangan umur yang berbeda untuk mengeksplorasi komputer, belajar pemrograman, dan untuk pekerjaan-pekerjaan yang biasa dilakukan dengan komputer yang lebih besar. Raspberry Pi bisa digunakan dalam pembelajaran pemrograman Python seperti untuk pembuatan aplikasi-aplikasi ataupun pemrograman bidang sistem otomatisasi. Selain itu, Raspberry Pi bisa digunakan untuk membuka aplikasi perkantoran, penampil gambar atau pembuatan *frame* foto, pemutar video, *game* komputer, dan lain-lain seperti kegunaan komputer pada umumnya.

Pada perkembangannya, Raspberry Pi telah dibuat dalam beberapa versi diantaranya yaitu versi A+, B+, 2B, dan 3B. Raspberry Pi 3B adalah versi terbaru dari Raspberry Pi yang dikeluarkan pada bulan Februari 2016. Secara fisik, perbedaan antara Raspberry Pi 2 dan Raspberry Pi 3 tidak terlalu terlihat. Perbedaan yang lebih menonjol adalah pada penambahan Wi-Fi dan Bluetooth pada Raspberry Pi 3 sehingga tidak memerlukan *adaptor* eksternal. Selain itu, dari segi CPU Raspberry Pi 3 menggunakan processor quad-core 1.2 GHz lebih baik dibanding Raspberry Pi 2 yang menggunakan quad-core 900MHz.

Pada Raspberry Pi 3 terdapat beberapa *port* yang tersedia seperti yang ditunjukkan pada Gambar 2.8.



Gambar 2.8. Raspberry Pi 3

- 4 pin USB
- 1 port Ethernet
- 1 port audio
- 1 port Kamera
- 1 port HDMI
- 1 micro USB untuk sumber daya
- 1 port micro SD
- 1 port display
- 40 pin GPIO (General Pin Input-Output)

Pada sistem absensi ini, Raspberry Pi dipilih sebagai kontroler dengan beberapa pertimbangan diantaranya penyimpanan yang lebih besar, mampu melakukan tugas-tugas seperti halnya komputer sehingga bisa dengan mudah dijadikan *file server*, terhubung ke internet karena memiliki wifi *adapter* ataupun dengan wifi *dongle*, dan mampu menulis, membaca, dan menyimpan *file excel* dan *CSV*.

2.5. Database

Database adalah sekumpulan tabel yang saling berhubungan yang ditunjukkan dengan kunci dari tiap tabel yang ada dan satu *database* menunjukkan sekumpulan data yang dipakai dalam satu lingkup instansi^[8]. Pada dasarnya *database* hanya sekedar informasi yang ada dan bisa diakses dalam periode waktu

yang lama. Dalam Bahasa umum, istilah *database* merujuk pada sekumpulan data yang dikelola oleh *Database Management System* (DBMS)^[9]. DBMS adalah suatu sistem yang dirancang untuk mengelola *database* dan menjalankan operasi pada *database*.

DBMS memiliki beberapa kemampuan diantaranya adalah memungkinkan pengguna untuk membuat *database* baru, menentukan skema data dan memodifikasi data. *Database* baru dibuat dengan menggunakan *Data Definition Language* (DDL) sedangkan modifikasi data dilakukan dengan menggunakan *Data Manipulation Language* (DML). DDL adalah sekumpulan perintah yang digunakan untuk membuat tabel, membuat index, mengubah struktur tabel, memberikan hak akses, dan lain sebagainya. Sedangkan DML adalah sekumpulan perintah untuk mengolah data seperti menampilkan data dari tabel, menyisipkan data kedalam tabel, menghapus data dari tabel dan lain sebagainya.

DBMS membantu untuk penyimpanan data dalam jumlah besar dengan rentang waktu yang lama. Data yang tersimpan tidak dapat dimodifikasi oleh pengguna yang tidak memiliki izin akses. Data tersebut dapat diakses oleh beberapa pengguna secara bersamaan dengan kecepatan akses data dan respon data yang tinggi. Walaupun diakses secara bersamaan, DBMS dapat menyeimbangkan konflik kebutuhan sehingga menghindari kerusakan data. Dengan demikian DBMS menjaga data tetap tersimpan dengan aman dan disimpan dalam waktu yang lama.

Terdapat beberapa komponen penting DBMS yaitu *storage manager*, *query processor*, dan *transaction manager*^[7]. *Storage manager* adalah komponen yang mengelola ruang penyimpanan data di dalam *disk* dan juga struktur data yang digunakan untuk merepresentasikan informasi yang tersimpan pada *disk*. *Query processor* adalah komponen yang berfungsi untuk menerjemahkan perintah dalam Bahasa query ke instruksi yang dimengerti oleh *database manager* untuk eksekusi data yang tersimpan. Sedangkan *transaction manager* adalah komponen untuk merekam perubahan pada *database* untuk membantu pemulihan setelah kerusakan sistem dan membantu mengeksekusi transaksi secara bersamaan.

2.6. Python

Python adalah bahasa pemrograman yang paling populer saat ini. Bahasa python adalah Bahasa level tinggi yang dibuat oleh Guido van Rossum pada tahun

1990 yang bisa digunakan untuk program prosedural dan program berbasis objek^[10]. Python digunakan dalam banyak bidang pemrograman seperti pengembangan web, pengembangan *software*, matematika, dan *scripting*. Bahasa pemrograman ini memiliki banyak kesamaan dengan Bahasa Inggris dan menggunakan indentasi untuk menentukan cakupan sebuah *looping*, fungsi, dan kelas dimana pada pemrograman umumnya menggunakan tanda kurung. Hal ini membuat Python menjadi Bahasa yang lebih muda untuk dibaca dan dimengerti.

Bahasa pemrograman Python dapat diperoleh dan digunakan secara bebas hingga bisa digunakan untuk kepentingan komersial. Pada saat ini distribusi Python sudah mencapai versi 3.8 yang dirilis pada 14 oktober 2019. Python bisa digunakan pada berbagai macam sistem operasi diantaranya Linux/Unix, Windows, Mac OS X, Java Virtual Machine, OS/2, Amiga, Palm, dan Symbian.

Sebagai Bahasa pemrograman tingkat tinggi, Python memiliki beberapa fitur diantaranya adalah sebagai berikut.

- Python memiliki *library* yang banyak sehingga tersedia modul-modul yang siap digunakan untuk berbagai macam keperluan.
- Tata Bahasa yang mudah dipahami karena banyak kesamaan dengan Bahasa Inggris.
- Memiliki aturan penulisan kode yang mudah untuk dibaca, dimengerti, maupun ditulis ulang.
- Bisa digunakan untuk pemrograman berbasis objek maupun pemrograman prosedural.
- Memiliki pengelolaan memori yang otomatis
- Mudah untuk menciptakan *library-library* baru baik menggunakan Bahasa Python sendiri maupun C atau C++.
- Memiliki banyak fasilitas pendukung sehingga memudahkan dalam pengoperasian.

Selain fitur-fitur tersebut masih banyak fitur lainnya yang menjadikan python sebagai Bahasa pemrograman yang populer.