

**PEMBUATAN PROYEK FINGERPRINT MENGGUNAKAN BAHASA
PEMROGRAMAN PYTHON DI CV. GIRISA TEKNOLOGI MALANG**

PRAKTIK KERJA LAPANGAN



**UNIVERSITAS
Ma CHUNG**

CHRISTOFER GENTA KRESNAMURTI SIMATUPANG

NIM : 311810009

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS MA CHUNG
MALANG
2021**

**LEMBAR PENGESAHAN
PRAKTIK KERJA LAPANGAN**

**PEMBUATAN PROYEK FINGERPRINT MENGGUNAKAN BAHASA
PEMROGRAMAN PYTHON DI CV. GIRISA TEKNOLOGI MALANG**

Oleh:
CHRISTOFER GENTA KRESNAMURTI SIMATUPANG
NIM. 311810009

dari:
PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS MA CHUNG

Dosen Pembimbing,

Mochamad Subianto, S.Kom., M.Cs.
NIP. 20100002

Dekan Fakultas Sains dan Teknologi

Dr. Kestrilia Rega Prilianti, M.Si.
NIP. 20120035

KATA PENGANTAR

Puji Syukur dan limpah terima kasih kepada Tuhan Yang Maha Esa karena telah memberikan hidayah-Nya sehingga penelitian ini dapat berjalan dengan lancar. Laporan ini disusun untuk memenuhi salah satu persyaratan dalam menyelesaikan PKL (Praktik Kerja Lapangan) bagi Mahasiswa dari Prodi Teknik Informatika Universitas Ma Chung Malang.

Atas dukungan moral dan materi yang telah diberikan dalam penyusunan laporan ini, penulis menyampaikan ucapan terima kasih kepada semua pihak yang telah membantu terutama kepada:

1. Orang tua yang telah mendukung dalam pelaksanaan praktik kerja lapangan ini.
2. Bapak Mochamad Subianto, S.Kom., M.Cs. Dosen prodi Teknik Informatika selaku pembimbing praktik kerja lapangan.
3. CV. Girisa Teknologi Malang yang telah memberikan kesempatan untuk melakukan praktik kerja lapangan.
4. Universitas Ma Chung Malang yang telah memberikan kesempatan untuk melakukan praktik kerja lapangan.
5. Saudara dan teman-teman yang telah memberikan dukungan moral.

Akhir kata, penulis berharap Tuhan YME membalas segala kebaikan semua pihak yang telah membantu. Semoga laporan Praktik Kerja Lapangan ini dapat bermanfaat dan menambah ilmu pembaca.

Malang , Desember 2021

Christofer Genta Kresnamurti Simatupang

DAFTAR ISI

Bab I	1
Pendahuluan	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	2
1.3 Batasan Masalah	2
1.4 Rumusan Masalah	2
1.5 Tujuan	3
1.6 Manfaat	3
Bab II	4
Gambaran Umum Perusahaan	4
2.1 CV. Girisa Teknologi	4
Bab III	7
Tinjauan Pustaka	7
3.1 Python	7
3.2 FAP10	8
3.3 Kabel Sensor	9
3.4 SQLYog	10
3.5 Tkinter	10
3.6 Pyfingerprint	11
3.7 Datetime	11
3.8 XAMPP	11
Bab IV	12
Perancangan Aplikasi dan Hasil	12
4.1 Perancangan Aplikasi	12
4.2 Proses Penggerjaan	14
4.3 Pengujian Program	23
4.4 Hasil	24
4.5 Diskusi	28
Bab V	29
Penutup	29

5.1	Simpulan	29
5.2	Saran	29
DAFTAR PUSTAKA		30

DAFTAR GAMBAR

Gambar 2.1 Struktur Organisasi CV. Girisa Teknologi	5
Gambar 2.2. Peta CV. Girisa Teknologi Malang	6
Gambar 3.1 Sensor FAP10 (Sumber: midastouchinc.com)	9
Gambar 3.2 Converter Penyambung Sensor FAP10	9
Gambar 3.3 Kabel Sensor (Sumber: learn.adafruit.com)	10
Gambar 4.1 Desain Database	13
Gambar 4.2 Alur Pembuatan Aplikasi	14
Gambar 4.3 Code Menghubungkan Program ke Database	15
Gambar 4.4 Code Pendaftaran Sidik Jari	16
Gambar 4.5 Code Lanjutan Pendaftaran Sidik Jari	18
Gambar 4.6 Code Absen Masuk Kerja	19
Gambar 4.7 Code Lanjutan Absen Masuk Kerja	20
Gambar 4.8 Code Absen Pulang Kerja	21
Gambar 4.9 Code Interface	22
Gambar 4.10 Proses Penyambungan Sensor Fingerprint	24
Gambar 4.11 Tampilan Aplikasi Fingerprint	25
Gambar 4.12 Pengujian Daftar Fingerprint	26
Gambar 4.13 Tampilan Database Sidik Jari	26
Gambar 4.14 Pengujian Absen Masuk	27
Gambar 4.15 Pengujian Absen Pulang	27
Gambar 4.16 Tampilan Database Absensi	27

Bab I

Pendahuluan

1.1 Latar Belakang

Teknologi merupakan suatu hal penting yang ada di dunia yang serba modern ini. Dengan berkembangnya zaman, semakin berkembang juga teknologi yang ada di dunia ini. Banyak sekali manfaat yang bisa kita rasakan dengan semakin berkembangnya teknologi. Tidak lain adalah bagaimana sekarang teknologi memudahkan para pengguna atau masyarakat yang ada di dunia ini untuk melakukan kegiatannya dan masih banyak lagi manfaat yang bisa kita dapat dari berkembangnya teknologi. Sehingga dengan semakin berkembangnya teknologi semakin mudah juga teknologi memecahkan masalah yang kompleks.

Setiap organisasi seperti halnya pada sekolah sangat membutuhkan sistem yang terkomputerisasi secara akurat, cepat, dan efisien (Safitri, Triyono, & Gunawan, 2018). Tidak hanya di sekolah, tetapi di suatu perusahaan pun juga membutuhkan sistem komputerisasi juga. Seperti contohnya adalah penggunaan alat *fingerprint* yang bertujuan untuk melakukan absensi pegawai. Saat ini memang banyak perusahaan yang sudah menggunakan *fingerprint* untuk melakukan absensi, tetapi masih banyak juga yang belum tahu jika penggunaan *fingerprint* ini bisa menggunakan bahasa pemrograman Python.

Kebanyakan masalah yang terjadi pada zaman sekarang dalam penggunaan *fingerprint* adalah kurang banyaknya variasi bahasa pemrograman untuk membuat sebuah program *fingerprint* seperti penelitian mahasiswa teknik informatika Universitas Wahid Hasyim mengenai perancangan sistem keamanan aliran arus listrik dengan *fingerprint* menggunakan arduino nano (Fajri & Mauludin, 2019). *Fingerprint* yang ada di pasaran saat ini juga tidak memiliki sistem untuk pengolahan data lebih lanjut, sehingga program hanya bisa melakukan absensi saja lalu dimasukkan ke dalam *cloud database* sehingga data tidak bisa diolah.

Penggunaan bahasa pemrograman Python sebagai *coding* untuk alat *fingerprint* FAP10 masih belum banyak digunakan karena bahasa pemrograman Python lebih sering digunakan sebagai bahasa *machine learning* dan *deep learning*,

hanya beberapa peneliti yang menggunakan Python sebagai bahasa pemrograman untuk *fingerprint* seperti merancang sistem presensi berbasis IoT yang menggunakan bahasa pemrograman Python (Pella, Likadja, Odja, & Ina, 2020). Bahasa Python sendiri memiliki tingkat kemudahan yang cukup mudah karena Python sendiri merupakan *coding* yang menggunakan bahasa manusia bukan komputer. Dan semakin berkembangnya zaman, Python sekarang banyak digunakan di semua aplikasi-aplikasi karena bahasanya yang mudah.

Karena hal tersebut dapat kita simpulkan, jika penggunaan bahasa pemrograman Python sebagai basis alat *fingerprint* FAP10 dapat dilakukan. Dengan adanya alat *fingerprint* FAP10 berbasis Python ini di CV. Girisa Teknologi diharapkan lebih banyak lagi aplikasi-aplikasi yang menggunakan berbasis Python.

1.2 Identifikasi Masalah

Berdasarkan latar belakang, ditemukan permasalahan sebagai berikut:

- a. Perlu adanya sistem absensi fingerprint menggunakan bahasa pemrograman python .
- b. Perlu adanya program fingerprint absensi sebagai alat absensi untuk karyawan.

1.3 Batasan Masalah

Adapun batasan masalah pada penelitian ini yaitu:

- a. Bahasa pemrograman yang digunakan Python.
- b. Alat fingerprint yang digunakan FAP10.

1.4 Rumusan Masalah

Rumusan masalah yang diambil adalah bagaimana membuat alat *fingerprint* FAP10 dengan menggunakan bahasa Python di CV Girisa Teknologi sehingga langsung terintegrasi dengan pusat data yang bisa pergunakan untuk keperluan selanjutnya

1.5 Tujuan

Tujuan dari PKL ini adalah mempelajari dan menghasilkan program dengan penggunaan Python sebagai bahasa pemrograman fingerprint FAP10 di CV Girisa Teknologi untuk membantu kinerja karyawan dan dapat diterapkan dalam kegiatan sehari-hari di CV. Girisa Teknologi Malang.

1.6 Manfaat

Manfaat yang diperoleh dari penelitian adalah sebagai berikut:

a. Bagi CV Girisa Teknologi

Dengan dibuatnya alat ini, diharapkan dapat mempermudah karyawan melakukan absensi di CV Girisa Teknologi.

b. Bagi Mahasiswa

Menambah pengetahuan penulis dan juga pengalaman dalam membuat aplikasi berbasis Python dan juga membantu penulis dalam menambah informasi dan referensi dalam pembuatan laporan.

c. Bagi Universitas Ma Chung

Dengan dibuatnya laporan ini, diharapkan dapat menambah bahan bacaan dan belajar untuk mahasiswa Program Studi Teknik Informatik

d. Bagi Peneliti

Dengan dibuatnya laporan ini, diharapkan dapat membantu peneliti selanjutnya untuk menambah informasi dan dapat membuat aplikasi berbasis python.

Bab II

Gambaran Umum Perusahaan

2.1 CV. Girisa Teknologi

CV. Girisa Teknologi adalah perusahaan yang bergerak dibidang Teknologi Informasi yang melayani jasa desain web, perancangan perangkat lunak, dan konsultasi seputar Teknologi Informasi. CV. Girisa Teknologi didirikan oleh Firmantya Adinugroho dan Joko Hari Prasetyo. Perusahaan ini didirikan pada tanggal 22 Januari 2013. Pada tanggal tersebut mereka meresmikan berdirinya perusahaan ini dihadapan Notaris Dian Trisnawan,S.H, M.Kn yang bertempat di Kota Malang dengan akte notaris No.97.

Perusahaan CV. Girisa Teknologi memanfaatkan perkembangan teknologi ke dalam dunia bisnis. Untuk memberikan pelayanan yang terbaik, CV. Girisa Teknologi memiliki visi dan misi untuk memberikan hasil kerja sesuai standar dengan CV. Girisa Teknologi itu sendiri. Berikut adalah visi dari CV. Girisa Teknologi:

a. Pelanggan

Mengerti akan kebutuhan pelanggan, memberi solusi serta pelayanan terbaik dan profesional kepada pelanggan.

b. Inovasi

Selalu melakukan inovasi dengan memanfaatkan kompetensi dan teknologi terikini, serta melakukan terobosan dalam proses kerja agar menjadi efisien dan efektif.

c. Kompeten

Mengembangkan sikap individu yang dapat diandalkan dan memiliki kompetensi yang sesuai dengan standar dan selalu mampu beradaptasi dengan teknologi terkini.

d. Integritas

Mengutamakan kejujuran, transparasi dan konsistensi antara pikiran, perkataan, dan perbuatan.

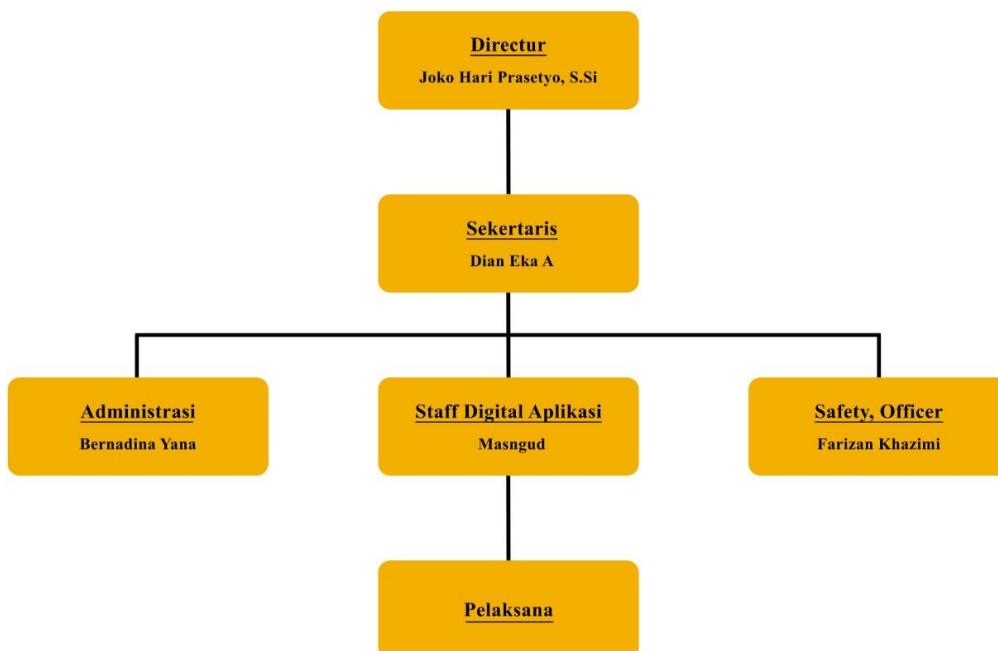
e. Kerjasama

Bekerjasama untuk mencapai tujuan perusahaan melalui sinergi berdasarkan prinsip saling percaya dan berbagi pengetahuan.

Sedangkan untuk misi dari CV. Girisa Teknologi yang bertujuan untuk bisa melakukan visinya adalah sebagai berikut:

- a. Mengembangkan produk-produk TI yang berkualitas dan memiliki daya saing yang kompetitif..
- b. Mengembangkan inovasi-inovasi teknologi terbaik dan terkini dalam setiap layanan dan produk.
- c. Memberikan layanan yang profesional serta membangun *long-term relationship* dengan customer dan *stake holder*.
- d. Menyelenggarakan kegiatan-kegiatan dalam bidang teknologi informasi dalam skala Nasional maupun Internasional.
- e. Memberikan kontribusi positif dalam perkembangan teknologi informasi Nasional maupun Internasional.

Struktur organisasi dari CV. Girisa teknologi dapat dilihat pada gambar berikut:



Gambar 2.1 Struktur Organisasi CV. Girisa Teknologi

Di dalam praktik kerja lapangan, penulis ditempatkan di Software Developer dengan pembimbing lapangan Bapak Masngud yang merupakan Staff Digital Aplikasi.

CV. Girisa Teknologi terletak di JL. Puncak Borobudur, No. 1, Kavling 33, Ruko Taman Borobudur Indah,, Mojolangu, Kec. Lowokwaru, Kota Malang. Peta lokasi CV. Girisa Teknologi dapat dilihat pada Gambar 2.2.



Gambar 2.2. Peta CV. Girisa Teknologi Malang

Bab III

Tinjauan Pustaka

3.1 Python

Python adalah bahasa pemrograman intepretatif yang dapat digunakan di berbagai platform dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode dan merupakan salah satu bahasa populer yang berkaitan dengan Data Science, Machine Learning, dan Internet of Things (IoT). Python sendiri menampilkan fitur-fitur menarik sehingga layak untuk dipelajari. Pertama, Python memiliki tata bahasa dan script yang sangat mudah untuk dipelajari. Python juga memiliki sistem pengelolaan data dan memori otomatis. Selain itu modul pada Python selalu diupdate. Ditambah lagi, Python juga memiliki banyak fasilitas pendukung. Python banyak diaplikasikan pada berbagai sistem operasi seperti Linux, Microsoft Windows, Mac OS, Android, Symbian OS, Amiga, Palm dan lain-lain.

Python dibuat dan dikembangkan oleh Guido Van Rossum, yaitu seorang programmer yang berasal dari Belanda. Pembuatannya berlangsung di kota Amsterdam, Belanda pada tahun 1990. Pada tahun 1995 Python dikembangkan lagi agar lebih kompatibel oleh Guido Van Rossum. Selanjutnya pada awal tahun 2000, terdapat pembaharuan versi Python hingga mencapai Versi 3 sampai saat ini. Pemilihan nama Python sendiri diambil dari sebuah acara televisi yang lumayan terkenal yang bernama Mothy Python Flying Circus yang merupakan acara sirkus favorit dari Guido van Rossum.

Banyak orang yang tertarik untuk menggunakan Python karena dianggap mudah untuk dipelajari, sekalipun oleh para pemula. Kode-kode yang ada didalamnya mudah dibaca dan dapat menjalankan banyak fungsi kompleks dengan mudah karena banyaknya standard library. Pengembangan program pada Python pun dapat dilakukan dengan cepat dan menggunakan lebih sedikit kode. Bahkan Python mampu menjadikan program dengan skala sangat rumit menjadi mudah. Python sendiri mendukung multi platform dan multi system serta memiliki sistem pengelolaan memori otomatis seperti Java.

Sayangnya Python cukup lambat dijalankan. Untuk pengembangan platform Android dan IOS juga terbilang kurang support. Python juga memiliki keterbatasan dengan akses basis data. Selain itu Python tidak cocok untuk melakukan tugas-tugas intensif memori dan pekerjaan multi-core/ multi-processor.

Dengan kode yang simpel dan mudah diimplementasikan, seorang programmer dapat lebih mengutamakan pengembangan aplikasi yang dibuat, bukan sibuk mencari syntax error.

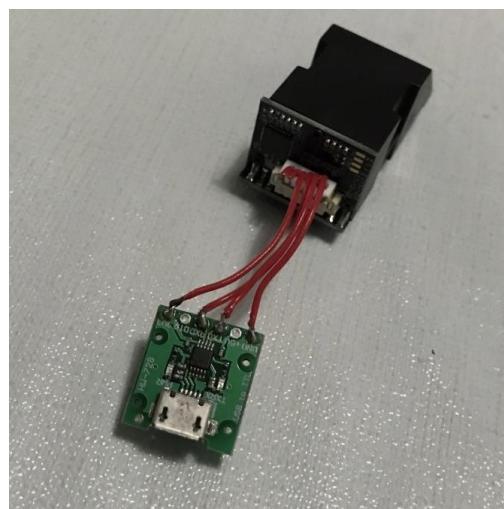
3.2 FAP10

FAP10 (*Fingerprint Acquisition Profile 10*) adalah sensor fingerprint yang tertanam MFC-1256 FBI yang tersertifikasi. FAP10 adalah sebuah sensor untuk akuisisi dan verifikasi sidik jari tingkat lanjut. Dengan resolusi 508 DPI dalam mengambil kualitas gambar yang bagus dengan akurasi membaca sidik jari yang tinggi dan juga hanya memakan waktu 0.07 detik untuk membaca dan memverifikasi sidik jari. FAP10 memiliki *built-in* analog dan digital sirkuit untuk meminimalkan jumlah komponen eksternal dan menyediakan tampilan antarmuka standart SPI yang mudah digunakan untuk mikroprosesor. Fungsi deteksi jari dapat mendeteksi jari secara otomatis dan menyalakan mikroprosesor dengan pin interupsi perangkat keras. Dan juga fungsi windowing dapat meotong gambar sidik jari yang lebih kecil dapat meningkatkan *frame rate* membaca gambar atau aplikasi lain.

Pada sensor fingerprint FAP10 ini dapat menyimpan 150 sidik jari di dalam memori sensor itu sendiri. Kabel sensor yang digunakan untuk memasang antara sensor FAP10 dengan micro converter hanya menggunakan 4 kabel yaitu gnd, RX, TX, dan VCC. Untuk kabel sensor gnd akan dipasangkan ke *board* gnd, sensor RX dipasangkan dengan *board* TX, sensor TX dipasangkan dengan *board* RX, dan sensor VCC dipasangkan dengan *board* 3.3v . Sensor FAP10 dapat disambungkan ke laptop ataupun komputer dengan menyambungkan kabel USB antara sensor FAP10 yang dipasangkan dengan port mikro dengan port USB yang ada pada komputer ataupun laptop.



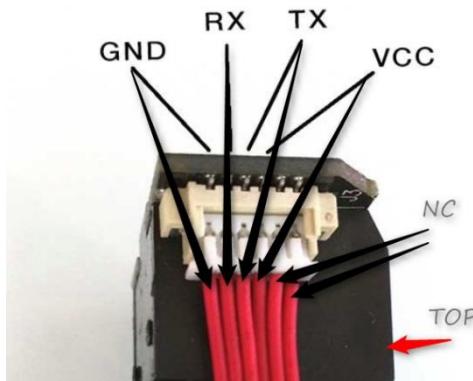
Gambar 3.1 Sensor FAP10 (Sumber: midastouchinc.com)



Gambar 3.2 Converter Penyambung Sensor FAP10

3.3 Kabel Sensor

Kabel sensor terdiri dari kabel-kabel yang berbeda. Kabel pertama yang paling kiri adalah *ground* (GND), tetapi jika memiliki kabel dengan warna yang berbeda-beda maka yang paling kiri adalah *black wire ground* (GND). Lalu dilanjutkan dengan 2 pin data, dan *power*



Gambar 3.3 Kabel Sensor (Sumber: learn.adafruit.com)

3.4 SQLYog

SQLYog merupakan aplikasi administrasi pada *Database* basis besar di Indonesia. SQLYog adalah aplikasi yang dikembangkan oleh Webyog yang berbasis di Kota Bangalore, India yang memiliki versi gratis atau *open source*, dan ada juga yang berbayar. SQLYog memiliki sangat beragam alat yang bisa membantu para penggunanya dalam mengelola *database*. Pada aplikasi ini dilengkapi dengan kelengkapan seperti *Procedure*, *Function*, dan *Trigger* dengan sangat mudah (N. E. Widianingsih, 2019).

3.5 Tkinter

Tkinter (Tk Interface) adalah suatu pustaka GUI (alat penghubung pemakai grafis) *widget* standar pembuatan Interface python untuk *Tk GUI Toolkit*. Tkinter merupakan antarmuka grafisnya dari TCL (Tool Command Language), yang memberikan kemudahan bagi para programmer dalam pembuatan programnya.

Tkinter merupakan pustaka grafis yang dapat memberikan kemudahan dalam pembuatan program berbasis grafis. Setiap GUI Toolkit menyediakan *widget*, yaitu objek user interface seperti *button*, *scrollbar*, *listbox*, *checkbox*, *radiobutton*, *label text*, dan lain sebagainya. *Widget* mengkapsulasi detil implementasi dan untuk setiap *widget* telah didefinisikan perilaku *default* sehingga mempermudah pemrograman GUI.

3.6 Pyfingerprint

Pyfingerprint adalah sebuah pustaka yang memungkinkan sensor sidik jari untuk melakukan deteksi sidik jari. Pada pustaka Pyfingerprint memberikan kemudahan untuk mengolah sidik jari seperti *enroll*, *search* sidik jari yang sudah di-*enroll*, *delete* sidik jari yang sudah di-*enroll*, *download* gambar hasil sidik jari, dan menghasilkan 32-bit nomer acak pada ZFM *hardware PRNG*.

3.7 Datetime

Datetime adalah sebuah pustaka atau modul yang dapat berfungsi menampilkan segala operasi yang berhubungan dengan waktu. Pada pustaka ini dapat menyesuaikan waktu seperti zona waktu. Terdapat 2 yaitu *aware* dan *naïve object*, untuk *aware object* menampilkan momen tertentu dalam waktu yang tidak terbuka untuk interpretasi. Sedangkan untuk *naïve object* tidak berisi informasi yang cukup untuk secara jelas terhadap objek tanggal/waktu lainnya (Python Software Foundation, n.d.).

3.8 XAMPP

XAMPP adalah sebuah perangkat lunak yang namanya diambil dari potongan beberapa kata seperti Apache, MySQL atau MariaDB, PHP, dan Perl. Sedangkan “X” yang berada di depan atau palaingkiri berasal dari sebuah *cross platform* yang untuk menjelaskan aplikasi ini dapat digunakan di berbagai macam sistem operasi seperti Windows, Linux, Mac, dan Solaris. XAMPP ini memiliki fungsi yang membuat server local dengan mengaktifkan beberapa modul dan biasanya digunakan dalam proses pengembangan *website*. Biasanya XAMPP juga digunakan untuk menguji aplikasi yang sedang dalam proses pengembangan ataupun menampilkan konten *website* tanpa harus terkoneksi dengan internet dan juga bisa membuat *web hosting* di local *server* tetapi hanya bisa diakses terbatas (Andy, 2019)

Bab IV

Perancangan Aplikasi dan Hasil

4.1 Perancangan Aplikasi

Sistem aplikasi yang akan dibangun adalah sebuah *project based* yang memiliki keluaran sebuah aplikasi desktop. Perencanaan yang akan dilakukan untuk mengerjakan aplikasi berikut adalah penentuan alur aplikasi, desain database, desain *user interface* (UI), dan hasil aplikasi.

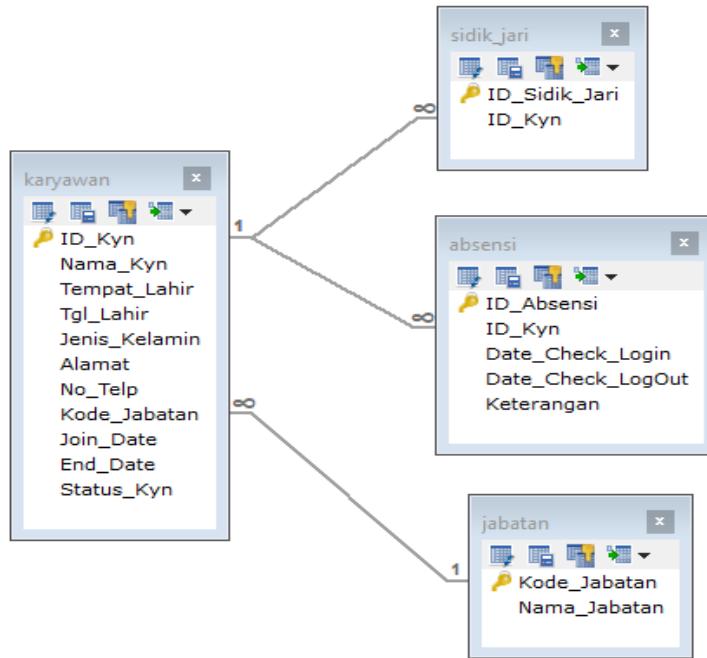
Menentukan alur aplikasi, desain database, dan desain *user-interface* (UI) diperlukan karena untuk mengetahui bagaimana jalannya aplikasi saat akan digunakan.

4.1.1 Alur Aplikasi

Tujuan utama untuk sistem yang akan dibangun adalah melakukan absensi. Untuk penyimpanan gambar hasil sidik jari akan disimpan di dalam sensor FAP10 sedangkan untuk ID-nya akan disimpan di dalam *database*. Saat menggunakan aplikasi untuk melakukan absensi, *user* harus melakukan daftar sidik jari dan memasukkan NIP terlebih dahulu melewati aplikasi dan lalu hasil dari pendaftaran tersebut akan dimasukkan ke dalam basis data absensi. Setelah terdaftar di dalam basis data absensi, *user* bisa melakukan absen untuk masuk dan pulang kerja dengan cara yang sama dengan saat melakukan daftar absensi, lalu pada aplikasi juga dapat melakukanhapus sidik jari.

4.1.2 Desain Database

Setelah menentukan alur aplikasi, dilanjutkan dengan membuat database. Desain database dari program fingerprint berikut dapat dilihat pada gambar yang terlampir berikut:



Gambar 4.1 Desain Database

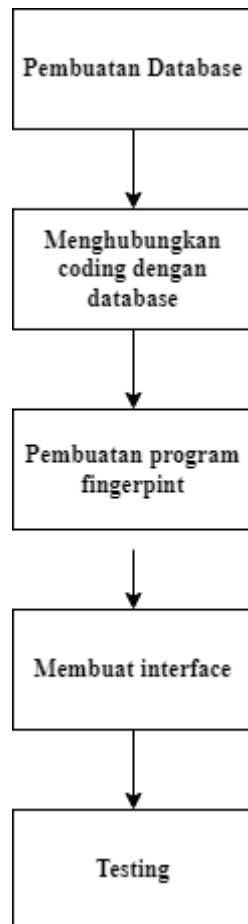
Pada database absensi ini, untuk penggunaan aplikasi fingerprint dengan menggunakan bahasa pemrograman python hanya menggunakan 3 dari 4 tabel yaitu karyawan, absensi, dan sidik jari. Dengan masing-masing fungsi yang berbeda seperti untuk tabel sidik jari akan menyimpan ID sidik jari dan juga ID karyawan (ID_Kyn) yang berhubungan dengan ID karyawan (ID_Kyn) yang ada di tabel karyawan. Lalu untuk tabel ketiga yaitu tabel absensi akan berfungsi untuk menyimpan data kehadiran karyawan dari jam masuk kerja dan juga jam pulang kerja.

4.1.3 Desain UI

Untuk desain *user interface* pada aplikasi ini cukup sederhana dan tidak rumit. Karena focus dari aplikasi ini adalah hanya untuk melakukan presensi maka aplikasi hanya akan mempunyai 5 tombol bersama *exit*. 4 tombol lainnya adalah untuk mendaftarkan sidik jari ke dalam *database*, lalu 2 tombol masing-masing untuk melakukan absen masuk kerja dan absen pulang, dan yang terakhir adalah tombol untuk menghapus sidik jari yang ada di dalam sensor *fingerprint* dan juga di *database*.

4.2 Proses Penggerjaan

Karena praktik kerja lapangan ini berbentuk *project based*, maka hasil keluaran yang diharapkan adalah aplikasi jadi. Dengan terbuatnya aplikasi ini diharapkan dapat membantu karyawan di tempat praktek kerja lapangan untuk melakukan absensi. Diagram alur dalam pembuatan aplikasi fingerprint ini seperti pada **Gambar 4.2** sebagai berikut:



Gambar 4.2 Alur Pembuatan Aplikasi

4.2.1 Pembuatan Database

Untuk pembuatan database absensi ini dikerjakan bersama dengan proyek tim yang melakukan PKL di CV. Girisa Teknologi Malang, sehingga akan memiliki kesinambungan dengan judul topik PKL dengan anggota yang lain.

Pada pembuatan database seperti yang sudah tertulis pada poin **4.1.2 Desain Database** akan sama dengan yang akan didesain dengan tabel yang sama.

4.2.2 Menghubungkan Program Dengan Database

Sesuai dengan desain rancangan aplikasi maka menghubungkan program dengan database adalah sebagai berikut:

```
1 from tkinter import *
2 from tkinter import messagebox
3 import time
4 import datetime
5 import hashlib
6 import tkinter
7 from tkinter import font
8 from pyfingerprint.pyfingerprint import PyFingerprint
9 from           pyfingerprint.pyfingerprint           import
10    FINGERPRINT_CHARBUFFER1
11    from           pyfingerprint.pyfingerprint           import
12      FINGERPRINT_CHARBUFFER2
13    import mysql.connector
14
15 mydb = mysql.connector.connect(
16   host= 'localhost',
17   username= 'root',
18   password= '',
19   database = 'absensi'
20 )
21
22 mycursor = mydb.cursor()
```

Gambar 4.3 Code Menghubungkan Program ke Database

Sebelum melakukan coding fingerprint, terlebih dahulu memanggil library apa saja yang akan dipakai seperti tkinter,tkinter messagebox, time, datetime, hashlib, pyfingerprint, dan mysql.connector. Memanggil library tkinter bertujuan untuk membuat interface aplikasi fingerprint dengan sederhana, sedangkan untuk time dan datetime memiliki tujuan yang sama yaitu menampilkan waktu secara real time saat melakukan absensi, untuk pyfingerprint memiliki fungsi yang penting dalam aplikasi ini yaitu untuk melakukan semua aktifitas fingerprint, dan mysql.connector berfungsi untuk menghubungkan program dengan database. Variabel mydb berfungsi sebagai penghubung program dengan database. Pada program ini akan menyambungkan program dengan database absensi dengan menggunakan XAMPP dan SQLyog.

4.2.3 Pembuatan Pendaftaran Sidik Jari Python

```

21 def enroll_finger():
22
23     try:
24         f      = PyFingerprint('COM4',      57600,      0xFFFFFFFF,
25                                0x00000000)
26
27         if ( f.verifyPassword() == False ):
28             raise ValueError('The given fingerprint sensor
password is wrong!')
29
30     except Exception as e:
31         print('The fingerprint sensor could not be
initialized!')
32         print('Exception message: ' + str(e))
33
34     try:
35         print('Waiting for finger...')
36
37         while ( f.readImage() == False ):
38             pass
39
40         f.convertImage(FINGERPRINT_CHARBUFFER1)
41
42
43         result = f.searchTemplate()
44         positionNumber = result[0]
45
46         if ( positionNumber >= 0 ):
47             print('Template already exists at position #' +
str(positionNumber))
48             exit(0)
49
50         print('Remove finger...')
51         time.sleep(2)
52
53         print('Waiting for same finger again...')
54
55         while ( f.readImage() == False ):
56             pass
57
58         f.convertImage(FINGERPRINT_CHARBUFFER2)
59
60         if ( f.compareCharacteristics() == 0 ):
61             raise Exception('Fingers do not match')
62
63         f.createTemplate()

```

Gambar 4.4 Code Pendaftaran Sidik Jari

Untuk pembuatan pendaftaran sidik jari unutk program ini hanya cukup memberikan 2 inputan yaitu pindaian sidik jari dan NIP karyawan. 2 inputan ini akan disimpan ke dalam database absensi di dalam tabel sidik jari.

Untuk isi coding berikut merupakan bagaimana cara untuk mendaftarkan sidik jari ke dalam database dengan menggunakan fungsi dari library Pyfingerprint lalu dimasukkan port tempat sensor fingerprint dimasukkan. Setelah itu untuk memeriksa apakah benar-benar sensorfingerprint terhubung atau tidak, menggunakan exception untuk mengirimkan pesan error yang terjadi seperti apa. Untuk penginisialisasi sensor fingerprint, untuk semua kelas pendaftaran, absen masuk dan pulang, dan juga penghapusan sidik jari memiliki kode yang sama seperti yang ada di atas.

Setelah sebelumnya melakukan inisialisasi sensor fingerprint, untuk coding berikut bertujuan untuk memindai sidik jari. Jika pindaian sidik jari berhasil maka hasil pindaian sidik jari sebelumnya akan diconvert ke dalam CHAR dan disimpan ke dalam FINGERPRINT_CHARBUFFER1 yang merupakan bawaan dari library pyfingerprint. Lalu setelah disimpan ke dalam FINGERPRINT_CHARBUFFER1, dalam variabel result memiliki fungsi untuk mencari apakah sidik jari sudah atau belum pernah terdaftar yang lalu disimpan ke dalam positionNumber dalam bentuk array result ke 0. Jika isi positionNumber ≥ 0 maka program akan keluar karena pindaian sidik jari sudah terdaftar ke dalam program, tetapi jika sidik jari masih belum terdaftar maka akan dilanjutkan dengan proses yang sama seperti sebelumnya yaitu memindai sidik jari lagi dan sebelumn melakukan sidik jari lagi, program akan memberikan perintah “*remove finger*” dan time sleep selama 2 detik yang berfungsi untuk memberi jeda sensor sebelum melakukan pindaian sidik jari yang kedua. Setelah melepas jari ke sensor, user melakukan pindaian yang kedua dengan proses yang sama seperti yang pertama tetapi untuk yang kedua kalinya hasil sidik jari akan disimpan ke dalam FINGERPRINT_CHARBUFFER2. Setelah itu program akan melakukan perbandingan antara hasil sidik jari yang tersimpan di dalam FINGERPRINT_CHARBUFFER1 dengan FINGERPRINT_CHARBUFFER2 apakah memiliki nilai 0 atau tidak. Jika disaat melakukan perbandingan didapat hasil 0 maka program akan mengeluarkan exception error dengan tulisan sidik jari

tidak sama, tetapi jika hasil perbandingan tidak memiliki nilai 0 maka program akan membuat template dari sidik jari yang sudah dibandingkan.

```
64      positionNumber = f.storeTemplate()
65      print('Finger enrolled successfully!')
66      print('New template position #' + 
str(positionNumber))
67      nip = input("Masukkan NIP: ")
68
69      sql= "INSERT INTO Karyawan (ID_Kyn) VALUES ('"+ 
nip+"')"
70      mycursor.execute(sql)
71
72      sql2= "INSERT INTO sidik_jari (ID_Kyn, 
ID_Sidik_Jari) VALUES ('"+ nip+"', '"+ 
str(positionNumber)+"')"
73      mycursor.execute(sql2)
74
75
76      mydb.commit()
77      messagebox.showinfo(message="1 record inserted, 
ID:" +nip)
78
79      except Exception as e:
80          print('Operation failed!')
81          print('Exception message: ' + str(e))
82          exit(1)
```

Gambar 4.5 Code Lanjutan Pendaftaran Sidik Jari

memberikan pesan jika program pendaftaran sidik jari memiliki eror. Dan jika terjadi eror maka program akan langsung tertutup.

4.2.4 Pembuatan Absensi Masuk Kerja Python

```

83 def searchmasuk_finger():
84
85     try:
86         f      =      PyFingerprint('COM4',      57600,      0xFFFFFFFF,
87                                     0x00000000)
88
89         if ( f.verifyPassword() == False ):
90             raise ValueError('The given fingerprint sensor
91 password is wrong!')
92
93     except Exception as e:
94         print('The fingerprint sensor could not be
95 initialized!')
96         print('Exception message: ' + str(e))
97         exit(1)
98
99     ## Gets some sensor information
100    print('Currently used templates: ' +
101          str(f.getTemplateCount()) +'/' + str(f.getStorageCapacity()))
102
103    ## Tries to search the finger and calculate hash
104    try:
105        print('Waiting for finger...')
106
107        ## Wait that finger is read
108        while ( f.readImage() == False ):
109            pass
110
111        ## Converts read image to characteristics and
112        ## stores it in charbuffer 1
113        f.convertImage(FINGERPRINT_CHARBUFFER1)
114
115        ## Searches template
116        result = f.searchTemplate()
117
118        positionNumber = result[0]
119
120        timenow= datetime.datetime.now()

```

Gambar 4.6 Code Absen Masuk Kerja

Pada coding class untuk absensi masuk memiliki awalnya yang sama dengan pendaftaran sidik jari seperti yang sudah ditulis pada sebelumnya, pertama akan melakukan inisialisasi sensor dengan port USB yang tersedia. Setelah melakukan pemindaian, program akan mencari informasi mengenai sidik jari nomor ke berapa sidik jari yang melakukan absensi dengan mencari template dan juga mengkalkulasi *hash*-nya. Setelah melakukan pencarian pada variabel result, lalu isi result yang ke-0 akan disimpan ke dalam variabel postionNumber. Lalu variabel timenow berfungsi untuk menampilkan waktu secara realtime.

```

115     if ( positionNumber == -1 ):
116             print('No match found!')
117             exit(0)
118         else:
119             if timenow > todayAt(8,5):
120
121                 a= "SELECT ID_Kyn FROM sidik_jari WHERE
122                 ID_Sidik_Jari=( '"+ str(positionNumber) + "' )"
123
124                 mycursor.execute(a)
125                 myresult = mycursor.fetchone()
126                 for x in myresult:
127                     sql2      = "INSERT      INTO      absensi
128                     (ID_Kyn,Date_Check_Login)      VALUES      ('"+str(x)+"', '"++
129                     str(timenow) + "' )"
130
131                     messagebox.showinfo(title="check",
132                         message="NIP anda adalah "+str(x)+"\n" "absen pada jam
133                         "+str(timenow) +"\n Anda Terlambat!")
134                     mycursor.execute(sql2)
135
136                 else:
137                     a= "SELECT ID_Kyn FROM sidik_jari WHERE
138                     ID_Sidik_Jari=( '"+ str(positionNumber) + "' )"
139
140                     mycursor.execute(a)
141                     myresult = mycursor.fetchone()
142                     for x in myresult:
143                     sql2      = "INSERT      INTO      absensi
144                     (ID_Kyn,Date_Check_Login)      VALUES      ('"+str(x)+"', '"++
145                     str(timenow) + "' )"
146
147                     messagebox.showinfo(title="check",
148                         message="NIP anda adalah "+str(x)+"\n" "absen pada jam
149                         "+str(timenow) )
150                     mycursor.execute(sql2)
151
152                 mydb.commit()
153
154             except Exception as e:
155                 print('Operation failed!')
156                 print('Exception message: ' + str(e))
157                 exit(1)

```

Gambar 4.7 Code Lanjutan Absen Masuk Kerja

Lalu diperiksa apakah hasil pindaian sidik jari tadi memiliki isi positionNumber sama dengan -1 atau tidak. Jika memiliki nilai -1 maka pindaian sidik jari tidak ditemukan dan program akan menutup dengan sendirinya sedangkan jika tidak senilai -1 maka program akan berlanjut ke dalam kondisi dimana user melakukan absensi di atas pukul 08.05 dan dibawah pukul 08.05. Jika user melakukan absensi di atas pukul 08.05 maka program akan mengeluarkan messagebox keterangan jika user terlambat. Sebelumnya program akan mencari

ID_Kyn terlebih dahulu di dalam tabel sidik jari untuk mengetahui siapa yang sedang melakukan absensi dengan kueri SELECT yang tertulis pada variabel a. Lalu kueri dijalankan dan diambil hanya 1 row saja menggunakan fungsi fetchone karena hanya ada 1 ID sidik jari untuk setiap 1 ID karyawan. Setelah itu dilakukan pemeriksaaan pada hasil pengambil 1 row tadi, dan dilakukan INSERT INTO atau memasukkan ID_Kyn yang sudah dicari dengan menggunakan variabel a tadi dan waktu sekarang atau *real-time* ke dalam tabel absensi. Dan untuk user yang melakukan absensi di bawah pukul 08.05 memiliki proses yang sama dengan dengan saat user melakukan absensi di atas pukul 08.05 tetapi hanya saja tidak menampilkan messabeox dengan pesan keterangan jika user terlambat. Dan pada exception sama dengan sebelumnya, program akan memberikan pesan eror yang terjadi apa dan dimana.

4.2.5 Pembuatan Absensi Pulang Kerja Fingerprint

```

146     if ( positionNumber == -1 ):
147             print('No match found!')
148             exit(0)
149         else:
150
151             a=   "SELECT    ID_Kyn    FROM    sidik_jari    WHERE
152                 ID_Sidik_Jari=( '"+ str(positionNumber) +"' )"
153                 tanggal = datetime.datetime.now().date()
154                 mycursor.execute(a)
155                 myresult = mycursor.fetchone()
156                 for x in myresult:
157                     sql2      =   "UPDATE      absensi      SET
158                         Date_Check_LogOut=( '"+str(timenow)+" ')
159                         ID_Kyn=( '"+str(x)+" ')
160                         DATE(Date_Check_Login)=( '"+str(tanggal)+" ') "
161
162                     messagebox.showinfo(title="check",
163                         message="NIP anda adalah "+str(x)+"\n pulang pada jam
164                         "+str(timenow))
165                     mycursor.execute(sql2)

```

Gambar 4.8 Code Absen Pulang Kerja

Untuk pembuatan absensi pulang kerja memiliki proses yang sama dengan absensi masuk kerja. Tetapi pada absensi pulang kerja memiliki perbedaan proses dalam memasukkan data ke dalam database. Di saat user melakukan absen pulang kerja, program akan melakukan UPDATE untuk memasukkan absen pulang kerja ke dalam kolom DATE_CHECK_LOGOUT tetapi dengan membandingkan

terlebih dahulu jika ID_Kyn sama dengan ID user yang melakukan absensi dan membandingkan apakah tanggal absen masuk kerja paginya dengan tanggal pada waktu secara *real-time*. Pembandingan antara tanggal absen masuk kerja saat di pagi hari dengan tanggal pada waktu secara *real-time* adalah agar hasil absen tidak terjadi duplikasi pada tanggal-tanggal sebelumnya.

4.2.6 Pembuatan Interface Python

Sebelum membuat interface, terdapat fungsi TodayAt yang berfungsi untuk membuat batasan waktu yang dibandingkan di saat karyawan melakukan absensi kerja, apakah terlambat atau tidak.

```
160     def todayAt(hr, min=0, sec=0, micros=0):
161         now = datetime.datetime.now()
162         return now.replace(hour=hr, minute=min, second=sec,
163                         microsecond=micros)
164
165     window = Tk()
166     window.geometry("450x450")
167
168     l1= Label(text="ABSENSI CV. GIRISA TEKNOLOGI", font=20)
169     l1.place(relx=0.5, rely=0.2, anchor=CENTER)
170
171     jam= Label(window, font=("Courier", 14, 'bold'), bd=14)
172     jam.place(relx=0.5, rely=0.9, anchor=CENTER)
173
174     def digitalclock():
175         text_input = time.strftime("%H:%M:%S")
176         jam.config(text=text_input)
177         jam.after(200, digitalclock)
178
179     a= Button(text="Daftar Fingerprint",fg="white", bg="blue",
180               command=enroll_finger)
181     b= Button(text="Absen Masuk",fg="white", bg="blue",
182               command=searchmasuk_finger)
183     c= Button(text="Absen Pulang",fg="white", bg="blue",
184               command= searchpulang_finger)
185     d= Button(text="Exit",fg="white", bg="blue", command=exit)
186
187
188     digitalclock()
189     window.mainloop()
```

Gambar 4.9 Code Interface

Pada pembuatan interface dengan menggunakan library tkinter, variabel window akan memanggil fungsi Tk dan mengatur ukuran sebesar 450 x 450 pixel. Dan variabel 11 berfungsi untuk memberikan label pada program dengan bertuliskan ABSENSI CV. GIRISA TEKNOLOGI dengan ukuran tulisan sebesar 20 dengan peletakkan pada sumbu x atau sumbu horizontal sebesar 0.5 atau berada di tengah-tengah tampilan aplikasi dan anchor CENTER. Sama seperti pada 11, variabel jam akan berada di posisi tengah bawah aplikasi yang bertujuan untuk menampilkan waktu secara *real-time* dengan class digitalclock dari library time yang mengambil waktu secara *real-time* yang lalu ditampilkan di interface aplikasi.

Setelah itu membuat button-button untuk melakukan daftar fingerprint, absen masuk, absen pulang, menghapus fingerprint, dan keluar dari program dengan tampilan tombol berwarna biru dan berubah putih ketika tombol ditekan. Lalu diberikan command sesuai dengan masing-masing kegunaan tombol. Setelah itu tombol akan terletak di tengah aplikasi dengan saling berurutan. Setelah itu memanggil fungsi digitalclock tadi untuk menampilkan waktu *real-time* dan dilakukan mainloop agar aplikasi tidak tertutup dengan sendirinya jika sudah melakukan 1 perintah.

4.3 Pengujian Program

Pengujian program fingerprint dengan menggunakan bahasa pemrograman python dilakukan dengan cara menyambungkan kabel data micro yang lalu disambungkan ke dalam sensor FAP10.



Gambar 4.10 Proses Penyambungan Sensor Fingerprint

4.4 Hasil

Hasil dari praktik kerja lapangan di CV. Girisa Teknologi Malang menghasilkan program aplikasi fingerprint dengan menggunakan bahasa pemrograman python. Hasil dan pengujian dari program ini diharapkan dapat membantu dan dapat diimplementasikan pada CV.Girisa Teknologi Malang.

4.4.1 Tampilan Aplikasi

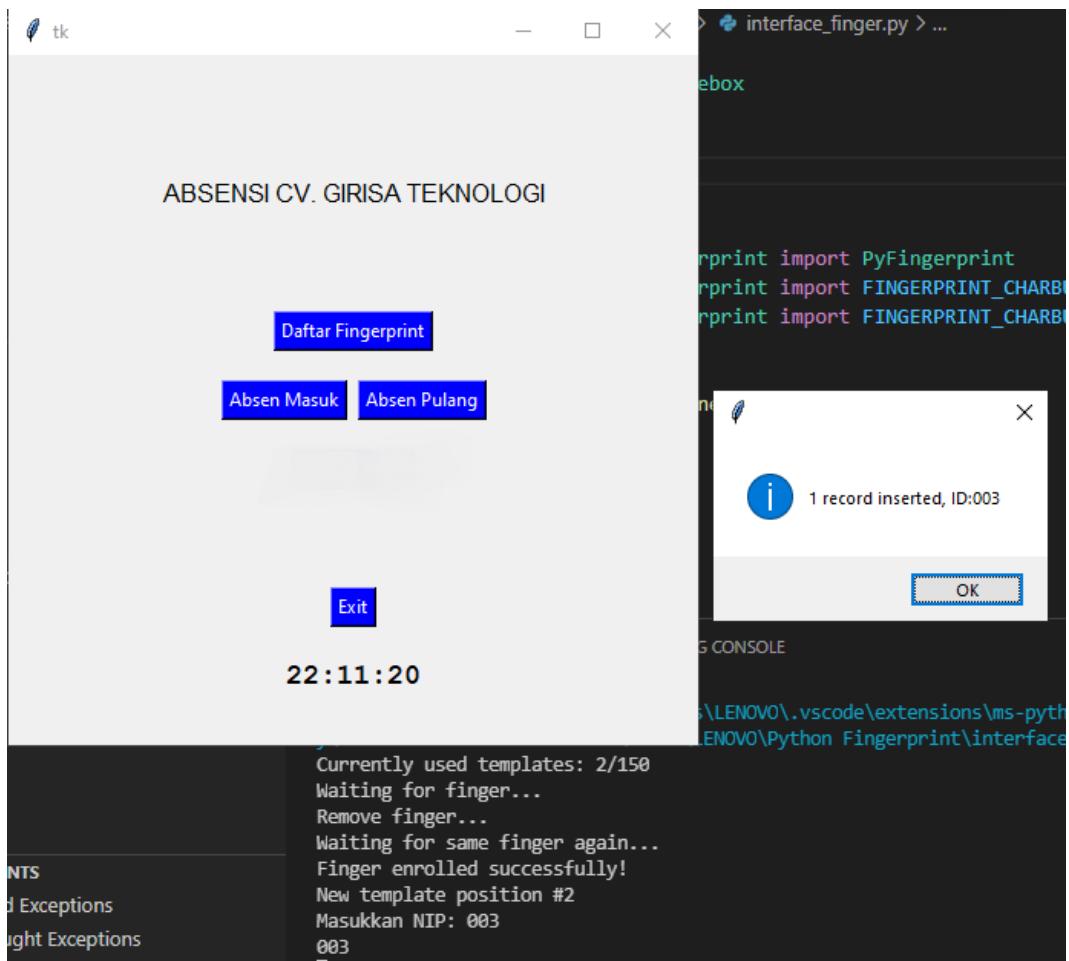
Untuk tampilan aplikasi cukup sederhana dengan hanya menampilkan 5 tombol yaitu daftar fingerprint, absen masuk, absen pulang, delete fingerprint, dan exit. Tampilan aplikasi seperti gambar berikut:



Gambar 4.11 Tampilan Aplikasi Fingerprint

4.4.2 Pendaftaran Fingerprint

Tampilan saat melakukan pendaftaran fingerprint akan ditampilkan di dalam terminal atau command prompt. Dan untuk menginput NIP juga berada di dalam terminal atau command prompt. Setelah menginputkan NIP maka program akan menampilkan message box.



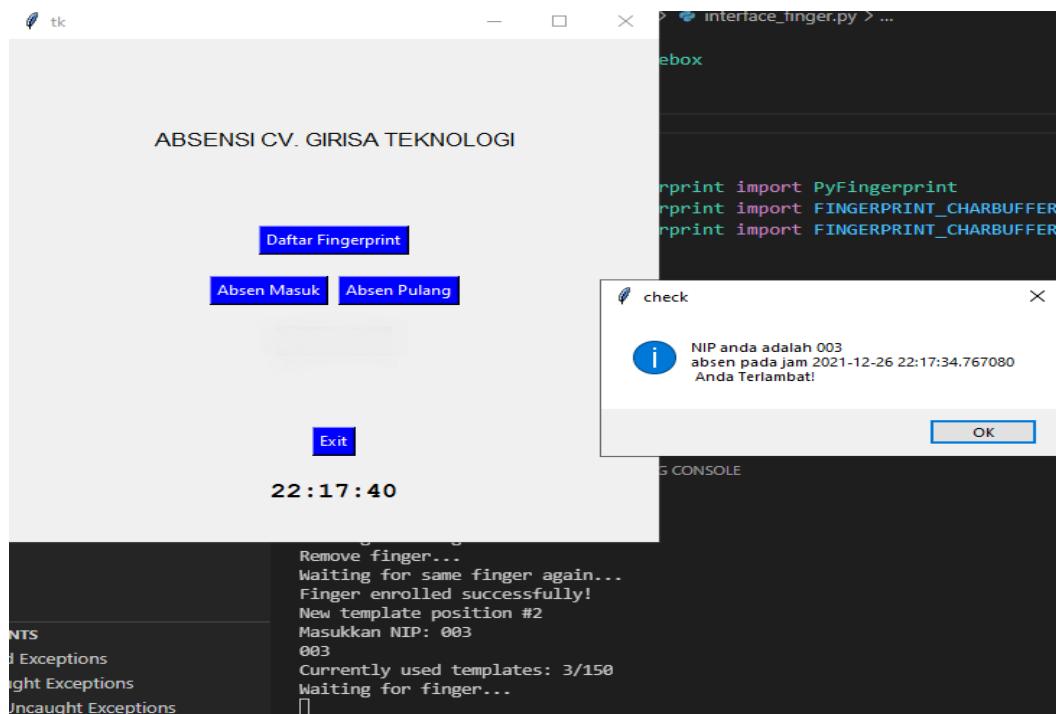
Gambar 4.12 Pengujian Daftar Fingerprint

	ID_Sidik_Jari	ID_Kyn
0	001	
1	002	
2	003	
*	(NULL)	(NULL)

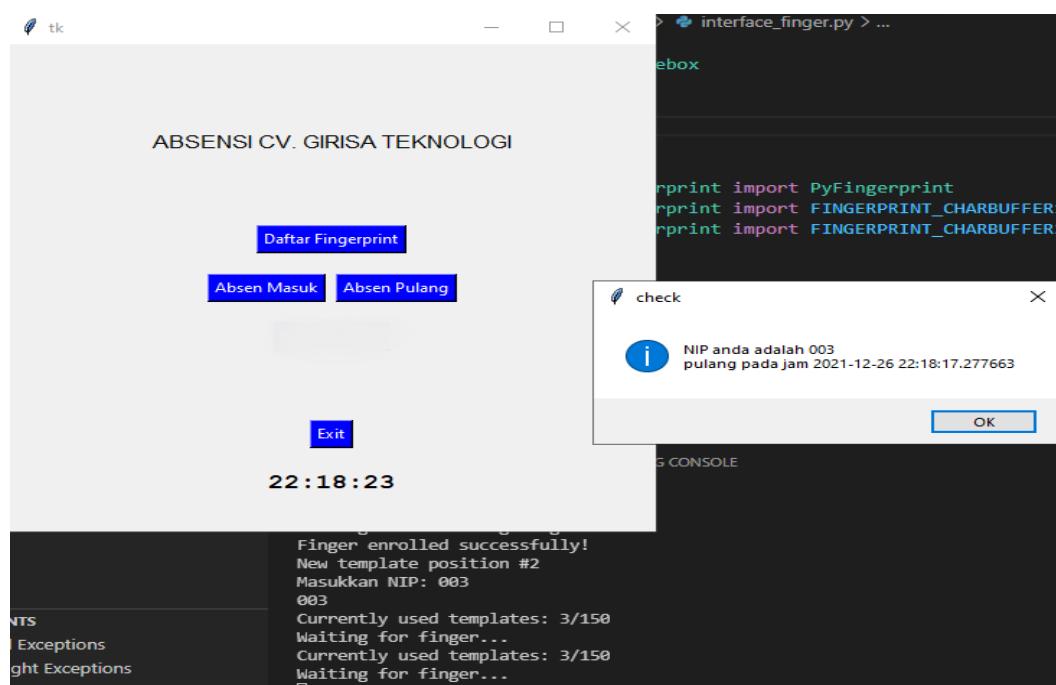
Gambar 4.13 Tampilan Database Sidik Jari

4.4.3 Absen Masuk dan Absen Pulang

Untuk absen masuk dan absen pulang memiliki tampilan yang sama tetapi memiliki message box yang berbeda. Setelah melakukan pemindaian sidik jari, data ID karyawan dan waktu absensi akan langsung terinput ke dalam database.



Gambar 4.14 Pengujian Absen Masuk



Gambar 4.15 Pengujian Absen Pulang

	ID_Absensi	ID_Kyn	Date_Check_Log	Date_Check_LogOut	Keterangan
□	38	001	2021-12-23 13:32:40	2021-12-23 14:45:51	(NULL)
□	39	003	2021-12-26 22:17:34	2021-12-26 22:18:17	(NULL)
*	(Auto)	(NULL)	(NULL)	(NULL)	(NULL)

Gambar 4.16 Tampilan Database Absensi

4.5 Diskusi

Pada praktik kerja lapangan yang sudah dilakukan penulis menghasilkan program fingerprint dengan mengembangkan program menggunakan bahasa pemrograman python berhasil dilaksanakan dengan lancar. Program yang sudah dibuat dapat dikembangkan dan diterapkan di CV. Girisa Teknologi Malang dengan harapan program dapat membantu dalam pengimplementasian kinerja sehari-hari karyawan CV. Girisa Teknologi Malang.

Bab V

Penutup

5.1 Simpulan

Berdasarkan hasil praktik kerja lapangan yang sudah dilakukan di CV. Girisa Teknologi dapat disimpulkan bahwa program yang dikerjakan dapat berfungsi dengan sesuai harapan. Hal ini dapat dibuktikan dengan melihat hasil proses pengujian program yang dapat berjalan dan berfungsi dengan baik tanpa *error*. Program dapat dikatakan berfungsi dengan sesuai harapan dengan bukti sebagai berikut:

1. Ketika dilakukan uji program, program berjalan dengan baik tanpa *error* dan dapat mengeluarkan maupun menerima inputan dengan baik
2. Hasil pindaian sidik jari dan NIP dapat masuk ke dalam database
3. Program dapat melakukan semua fitur yang dibutuhkan yaitu pendaftaran sidik jari, absen masuk dan pulang kerja, dan menghapus sidik jari

5.2 Saran

Saran untuk pembuatan fingerprint dengan bahasa pemrograman *python* adalah:

1. Memahami terlebih dahulu jalan atau alur database terlebih dahulu lalu baru membuat programnya. Supaya tidak terjadi kebingungan di tengah jalan.
2. Memastikan port yang digunakan ketika menyambungkan sensor FAP10 dengan laptop maupun komputer. Karena jika port yang disambungkan di program tidak sama dengan port yang disambungkan ke perangkat keras maka program tidak akan bisa berjalan.

DAFTAR PUSTAKA

- Andy. (2019, September). *Pengertian XAMPP*. Retrieved from Q Words: <https://qwords.com/blog/pengertian-xampp/>
- Fajri, F. A., & Mauludin, M. S. (2019). RANCANG BANGUN SISTEM KEAMANAN ALIRAN LISTRIK ARUS AC dengan FINGERPRINT menggunakan ARDUINO NANO. *Informatika dan RPL Vol.2, No.1*, 26-31.
- Pella, S. I., Likadja, F., Odja, M., & Ina, W. T. (2020). Rancang Bangun Sistem Presensi Berbasis IoT. *Media Elektro*, 60-67.
- Python Software Foundation. (n.d.). *Python: datetime - Basic date and time types*. Retrieved from Python: <https://docs.python.org/3/library/datetime.html>
- Safitri, R., Triyono, & Gunawan, T. (2018). *PERANCANGAN SISTEM INFORMASI ABSENSI GURU DAN STAFF PADA SMK PANCA KARYA TANGERANG BERBASIS WEB*. Tangerang: Media Neliti.

LAMPIRAN

 UNIVERSITAS MA CHUNG	 FAKULTAS SAINS & TEKNOLOGI UNIVERSITAS MA CHUNG	<p> Villa Puncak Tidar N-01 Malang 65151 fakultas.sains.teknologi@machung.ac.id +62 341 550171  www.machung.ac.id </p> <div style="border: 1px solid black; padding: 2px; margin-top: 10px;"> FORM PKL-FST04 </div>
--	---	---

**LEMBAR PENILAIAN PRAKTIK KERJA LAPANGAN (PKL)
PIHAK EKSTERNAL**

Nama Mahasiswa	:	Christofer Genta K.S
NIM	:	311810009
Program Studi	:	Teknik Informatika
Pembimbing Lapangan	:	Bpk. Masngud
Tanggal Pelaksanaan	:	1 Juli 2021 – 23 Desember 2021
Tempat PKL	:	CV. Girisa Teknologi
Topik PKL	:	PEMBUATAN PROYEK FINGERPRINT MENGGUNAKAN BAHASA PEMROGRAMAN PYTHON DI CV. GIRISA TEKNOLOGI MALANG

Komponen Penilaian :

No	Macam Kegiatan yang dinilai	Nilai Angka (0-100)
1	Disiplin dalam kehadiran (presensi)	91
2	Kesungguhan dalam melakukan praktik kerja	92
3	Disiplin dalam pekerjaan	94
4	Kemampuan memecahkan masalah	95
5	Tanggung jawab	92
6	Kemauuan untuk mengetahui hal-hal yang ada di tempat praktik kerja	85
7	Pengetahuan tentang ilmu yang dilaksanakan	92
8	Keterampilan	94
9	Kemampuan menyampaikan pendapat	90
10	Kemampuan dalam bekerja sama	89
<i>Rerata (Jumlah Nilai : 10 komponen)</i>		91,4

Kamis, 23 December 2021

MUTU	RENTANG	BOBOT NILAI
A	95,00 s/d 100	4,00
AB	90,00 s/d 94,99	3,70
BA	82,00 s/d 89,99	3,30
B	73,00 s/d 81,99	3,00
BC	65,00 s/d 72,99	2,70
CB	60,00 s/d 64,99	2,30
C	56,00 s/d 59,99	2,00
CD	50,00 s/d 55,99	1,50
D	40,00 s/d 49,99	1,00
E	00,00 s/d 39,99	0,00

Pembimbing Lapangan*





JURNAL HARIAN PEMBIMBINGAN PRAKTIK KERJA LAPANGAN (PKL)

Nama Mahasiswa	:	Christofer Genta K.S
NIM	:	311810009
Program Studi	:	Teknik Informatika
Posisi PKL	:	Software Developer
Nama Instansi/Perusahaan	:	CV. Girisa Teknologi
Alamat Instansi/Perusahaan	:	JL. Puncak Borobudur, No. 1, Kavling 33, Ruko Taman Borobudur Indah,, Mojolangu, Kec. Lowokwaru, Kota Malang
Nama Pembimbing Lapangan	:	Bpk. Masngud

No	Hari, tanggal	Uraian Kegiatan PKL	TTD Dosen Pembimbing Lapangan
1	Rab, 1 Jul	Pemberian proyek progressive web app dan fingerprint	
2	Rab, 7 Jul	Konsultasi proyek yang dipilih melalui daring	
3	Kam, 10 Jul	Konsultasi proyek yang dipilih melalui daring	
4	Kam, 15 Jul	Memilih proyek website PWA untuk dikerjakan daring	
5	Kam, 9 Sep	Ganti proyek menuju fingerprint python	
6	Jum, 17 Sep	Mengambil hardware di CV. Girisa Teknologi	
7	Jum, 24 Sep	Konsultasi daring tentang koneksi fingerprint	
8	Kam, 7 Oct	Konsultasi daring cara mengubah fingerprint code menjadi python	
9	Rab, 13 Oct	Konsultasi tentang database fingerprint	
10	Sel, 19 Oct	Konsultasi penggantian hardware fingerprint	
11	Kam, 23 Dec	Pengumpulan dan penilaian	

LEMBAR BIMBINGAN PRAKTIK KERJA LAPANGAN DARING (PKL)

Nama Mahasiswa	:	Christofer Genta K.S
NIM	:	311810009
Program Studi	:	Teknik Informatika
Judul Laporan PKL	:	PEMBUATAN PROYEK FINGERPRINT MENGGUNAKAN BAHASA PEMROGRAMAN PYTHON DI CV. GIRISA TEKNOLOGI MALANG

No	Hari, tanggal	Topik Bimbingan	TTD Dosen Pembimbing
1	Selasa,12 Oktober 2021	Konsultasi fingerprint menggunakan database sql atau tidak	
2	Selasa,19 Oktober 2021	Konsultasi fingerprint awal diganti menjadi fingerprint FAP10	
3	Kamis,4 November 2021	Konsultasi pembuatan database	
4	Jumat,17 Desember 2021	Konsultasi mengirim ID karyawan dan ID sidik jari ke dalam database	
5	Senin,20 Desember 2021	Konsultasi kueri melakukan absensi dari python ke sqlyog	
6	Kamis,23 Desember 2021	Konsultasi perbaikan kueri dan menampilkan hasil absensi dari python ke sqlyog atau database	
7			
8			
9			
10			

