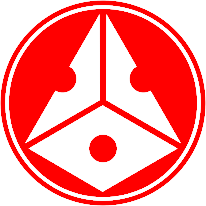
## LAPORAN TUGAS AKHIR MATA KULIAH MICROCONTROLLER

**“Sistem Absensi Berbasis Fingerprint Menggunakan Arduino Uno untuk perhitungan Gaji berdasarkan Absensi”**



Dosen Pengampu : Faisal Akbar, S.Kom., M. T.

Disusun Oleh :

|  |  |
| --- | --- |
| Rifky Hilmanudin  Anida Aulia Utami  Merliayana  Afif Naufal | 24523301  14522638  14522647  14520758 |

# TEKNIK INFORMATIKA

**STIKOM PolTek Cirebon**

Jln. Bridgen Darsono No.33, Kedawung, Cirebon, Jawa Barat 45153, Indonesia Telp : +62 231 4864 | Website : https://[www.stikompoltek.ac.id](http://www.stikompoltek.ac.id/)

# KATA PENGANTAR

Puji syukur kehadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga saya dapat menyelesaikan laporan tugas akhir mata kuliah Microcontroller ini tepat pada waktunya.

Adapun tujuan dari penulisan laporan akhir ini adalah untuk memenuhi tugas akhir pada mata kuliah Microcontroller. Selain itu, laporan ini juga bertujuan untuk mendalami pengetahuan dan keterampilan dalam bidang microcontroller, khususnya dalam penerapan teknologi Arduino.

Saya mengucapkan terima kasih kepada Bapak Faisal Akbar, M. T. selaku dosen mata kuliah Microcontroller yang telah memberikan tugas akhir ini sehingga dapat menambah pengetahuan dan wawasan sesuai dengan bidang studi yang saya tekuni. Saya juga mengucapkan terima kasih kepada semua pihak yang tidak dapat saya sebutkan semua, terima kasih atas bantuannya sehingga sehingga saya dapat menyelesaikan tugas ini.

Saya menyadari, bahwa laporan yang saya tulis ini masih jauh dari kata sempurna. Oleh karena itu, kritik dan saran yang membangun akan saya nantikan demi kesempurnaan laporan tugas akhir ini.

Kuningan, 28 Juli 2023

Penyusun

i

# DAFTAR ISI

[KATA PENGANTAR i](#_bookmark0)

[DAFTAR ISI ii](#_bookmark1)

[DAFTAR GAMBAR iv](#_bookmark2)

[BAB I PENDAHULUAN 1](#_bookmark3)

* 1. [Latar Belakang 1](#_bookmark4)
  2. Rumusan Masalah 5
  3. Batasan Masalah 5
  4. Tujuan 6
  5. Manfaat 6

[BAB II ANALISIS DAN PERANCANGAN SISTEM 7](#_bookmark9)

* 1. [Deskripsi Sistem 7](#_bookmark10)
     1. [Alat dan Bahan 7](#_bookmark11)
     2. [Rangkaian 8](#_bookmark12)
  2. [Cara Kerja Sistem 8](#_bookmark14)
  3. Dasar Teori 9
     1. [Arduino Uno](#_bookmark16) 9
     2. [Sensor Fingerprint AS608 10](#_bookmark18)
     3. [RTC (Real Time Clock) 11](#_bookmark20)
     4. [LCD 16x2 11](#_bookmark22)
     5. SD Card Module 12
     6. Breadboard 13
     7. Kabel Jumper 13
     8. Lampu LED 14
     9. Buzzer 15

2.3.10 Push Button 15

* 1. [Diagram Blok Sistem 16](#_bookmark26)
  2. [Flowchart 17](#_bookmark28)

[BAB III HASIL DAN PEMBAHASAN 18](#_bookmark30)

* 1. [Hasil 18](#_bookmark31)
  2. [Program 21](#_bookmark35)

iii

# DAFTAR GAMBAR

[Gambar 1 Rangkaian Fritzing 8](#_bookmark13)

[Gambar 2 Arduino Uno 9](#_bookmark17)

[Gambar 3 Sensor Fingerprint AS608 10](#_bookmark19)

[Gambar 4 RTC (Real Time Clock) 11](#_bookmark21)

[Gambar 5 LCD 16x2 11](#_bookmark23)

[Gambar 6 SD Card Module 12](#_bookmark25)

[Gambar 7 Breadboard 13](#_bookmark27)

[Gambar 8 Kabel Jumper 13](#_bookmark29)

[Gambar 9 Lampu LED 14](#_bookmark32)

[Gambar 10 Buzzer 15](#_bookmark33)

[Gambar 11 Push Button 15](#_bookmark34)

[Gambar 12 Diagram Blok Sistem 16](#_bookmark34)

[Gambar 13 Flowchart 17](#_bookmark34)

[Gambar 14 Tabel Fingerprint 18](#_bookmark34)

[Gambar 15 Tabel Absensi 19](#_bookmark34)

[Gambar 16 Tabel Perhitung Gaji 19](#_bookmark34)

iv

# BAB I PENDAHULUAN

## Latar Belakang

Dalam era digital saat ini, teknologi semakin berkembang pesat dan berpengaruh besar dalam kehidupan sehari-hari. Salah satu teknologi yang menjadi tulang punggung revolusi ini adalah mikrokontroler. Mikrokontroler merupakan komponen elektronik yang mengintegrasikan fungsi mikroprosesor, memori, dan 1ndustry dalam satu chip tunggal. Dengan ukurannya yang kecil dan kemampuannya yang tinggi, mikrokontroler telah menjadi bagian integral dari berbagai sistem elektronik dan *embedded systems*. Salah satu 1ndust yang mendorong popularitas mikrokontroler adalah kemampuannya untuk mengotomatisasi proses dan mengendalikan sistem secara efisien. Mikrokontroler dapat untuk menjalankan fungsi-fungsi tertentu, mengumpulkan data dari sensor-sensor eksternal, berkomunikasi dengan perangkat lain melalui 1ndustry komunikasi, dan merespons perintah-perintah yang diberikan. Kemampuan ini menjadikan mikrokontroler sangat berguna dalam berbagai aplikasi seperti otomasi 1ndustry, kendali perangkat rumah tangga, sistem keamanan, kendaraan terhubung, robotika, dan masih banyak lagi.

Arduino Uno adalah salah satu jenis papan mikrokontroler yang sangat populer dan sering digunakan dalam proyek-proyek elektronika. Latar belakang munculnya Arduino Uno berkaitan erat dengan perkembangan dan evolusi mikrokontroler serta upaya untuk membuat teknologi mikrokontroler lebih mudah diakses oleh semua orang. Pada tahun 2005, proyek Arduino dimulai oleh

1

sekelompok mahasiswa dan dosen di *Interaction Design Institute Ivrea* di Italia. Tujuan proyek ini adalah untuk menciptakan platform sederhana yang dapat diakses oleh pemula, seniman, desainer, dan siapa saja yang tertarik dengan bidang elektronika. Dengan demikian, Arduino Uno menjadi hasil dari upaya untuk menciptakan lingkungan pengembangan mikrokontroler yang ramah pengguna, mudah dipahami, dan dapat diakses oleh pemula.

Arduino Uno didasarkan pada mikrokontroler ATMega328P yang kuat, yang diproduksi oleh perusahaan Atmel (sekarang diakuisisi oleh *Microchip Technology*). Papan ini dilengkapi dengan antarmuka yang sederhana dan user- friendly, termasuk pin-pin *input/output* (I/O) yang mudah diakses, konektor USB, dan rangkaian daya yang sudah terintegrasi. Selain itu, Arduino Uno dilengkapi dengan lingkungan pengembangan terpadu (*Integrated Development Environment/IDE*) yang intuitif, berbasis bahasa pemrograman *Wiring* dan didukung oleh bahasa pemrograman C/C++. Kelebihan utama Arduino Uno adalah kemampuannya untuk mempermudah pengguna dalam mempelajari dan mengembangkan proyek elektronika. Arduino Uno menyediakan kerangka kerja yang jelas dan dokumentasi yang kaya, yang memudahkan pemula untuk memahami konsep dasar mikrokontroler dan memulai dengan cepat. Dalam waktu singkat, Arduino Uno menjadi populer di kalangan komunitas elektronika, seniman, dan penghobi yang ingin menerapkan mikrokontroler dalam proyek-proyek kreatif mereka.

Sistem absensi berbasis fingerprint untuk perhitungan gaji berdasarkan absensi merupakan salah satu aplikasi menarik yang memanfaatkan kemampuan mikrokontroler Arduino untuk mencatat kehadiran karyawan dan menghitung gaji

mereka secara otomatis. Ide dasar di balik sistem ini adalah untuk menciptakan sebuah sistem otomatis yang mampu mencatat kehadiran karyawan dengan akurasi tinggi dan efisiensi, serta menghitung gaji mereka berdasarkan data absensi tersebut. Pencatatan absensi berbasis fingerprint memiliki berbagai contoh serupa dalam berbagai industri, termasuk industri manufaktur, perkantoran, dan layanan publik. Misalnya, dalam industri manufaktur, sistem ini dapat digunakan untuk memastikan bahwa karyawan hadir dan bekerja sesuai dengan jadwal yang ditentukan. Dalam perkantoran, aplikasi ini dapat membantu mengelola absensi karyawan dengan lebih efektif dan efisien, serta mengurangi potensi kesalahan manusia dalam pencatatan kehadiran. Dalam layanan publik, sistem absensi berbasis fingerprint dapat digunakan untuk memastikan kehadiran petugas layanan dan menghitung gaji mereka berdasarkan jam kerja yang telah tercatat. Selain itu, penggunaan sistem absensi berbasis fingerprint dapat meningkatkan keamanan dan akurasi dalam pencatatan kehadiran, karena setiap karyawan memiliki sidik jari yang unik dan tidak dapat dipalsukan. Dengan demikian, sistem ini dapat mencegah terjadinya kecurangan dalam pencatatan kehadiran dan memastikan bahwa data yang tercatat adalah akurat dan dapat dipercaya.

Sebelum adanya sistem absensi berbasis fingerprint untuk perhitungan gaji berdasarkan absensi, pencatatan kehadiran karyawan seringkali dilakukan secara manual atau menggunakan mesin absensi yang mahal dan rumit. Pencatatan manual membutuhkan banyak tenaga kerja manusia dan cenderung kurang efisien serta rentan terhadap kesalahan. Di sisi lain, mesin absensi yang tersedia di pasaran seringkali memiliki biaya yang tinggi dan sulit dikustomisasi untuk kebutuhan spesifik. Sistem absensi berbasis fingerprint berbasis Arduino hadir sebagai

alternatif yang lebih terjangkau dan fleksibel. Arduino Uno sebagai papan mikrokontroler yang digunakan dalam proyek ini, memberikan kemudahan dalam pemrograman dan koneksi dengan berbagai komponen tambahan, seperti sensor sidik jari dan modul RTC. Sensor sidik jari yang digunakan dapat mendeteksi dan mengidentifikasi sidik jari karyawan, kemudian mengirimkan informasi tersebut ke Arduino Uno. Arduino kemudian menganalisis data sidik jari tersebut dan mencatat waktu kehadiran karyawan dengan akurat. Dengan data absensi yang telah tercatat, sistem ini juga dapat menghitung gaji karyawan berdasarkan jumlah kehadiran dan jam kerja yang tercatat. Penggunaan RTC (Real Time Clock) memastikan bahwa waktu yang dicatat adalah akurat dan tepat, sementara penggunaan SD card memungkinkan penyimpanan data absensi secara aman dan dapat diakses kapan saja. Dengan menggunakan teknologi fingerprint dan Arduino, sistem absensi ini dapat menjadi solusi efektif untuk manajemen kehadiran karyawan dan perhitungan gaji yang akurat dan efisien.

Dengan menggunakan sistem absensi berbasis fingerprint untuk perhitungan gaji berdasarkan absensi, proses pencatatan kehadiran karyawan dapat dilakukan secara otomatis, akurat, dan efisien. Selain itu, fleksibilitas dan kemudahan dalam mengatur kriteria kehadiran dan perhitungan gaji memungkinkan aplikasi ini digunakan dalam berbagai konteks dan industri. Dengan demikian, sistem absensi berbasis fingerprint berbasis Arduino memberikan solusi yang efektif, terjangkau, dan fleksibel untuk manajemen kehadiran karyawan dan perhitungan gaji, serta berpotensi untuk mengoptimalkan berbagai proses industri dan aplikasi lainnya.

## Rumusan Masalah

Berdasarkan Latar Belakang yang telah diuraikan maka, bisa dirumuskan masalah penelitan sebagai berikut :

1. Bagaimana merancang sistem absensi berbasis fingerprint yang dapat mencatat kehadiran karyawan dan perhitungan gaji secara otomatis secara akurat dan efisien?
2. Bagaimana mengintegrasikan sensor sidik jari dengan mikrokontroler Arduino uno untuk mendeteksi dan mengidentifikasi sidik jari karyawan?
3. Bagaimana memastikan bahwa data waktu kehadiran yang di catat oleh sistem adalah akurat dan tepat waktu?

## Batasan Masalah

Dalam melakukan pembuatan projek ini, kami memiliki beberapa masalah yang ingin dibatasi diantaraya adalah :

1. Proyek ini akan difokuskan pada pencatatan kehadiran karyawan berdasarkan sidik jari dengan perhitungan gaji yang dihitung berdasarkan data absensi tersebut. Batasan ini mempermudah dalam pengenalan dan pencatatan kehadiran, tetapi juga membatasi kemampuan untuk mengelola data kehadiran dengan aspek lain seperti cuti atau lembur.
2. Sistem ini dirancang untuk mencatat waktu masuk dan keluar karyawan serta menghitung jam kerja untuk perhitungan gaji, tetapi tidak mencakup fitur lain seperti manajemen cuti, lembur, atau izin. Batasan ini memungkinkan fokus pada pencatatan dan perhitungan dasar,

sementara aspek manajemen kehadiran yang lebih kompleks mungkin tidak tersedia.

1. Fitur lain seperti penyimpanan data kehadiran di server eksternal atau integrasi dengan sistem HR yang lebih luas belum tersedia. Batasan ini memungkinkan fokus pada pengembangan sistem lokal dengan penyimpanan data di SD card, sementara kemampuan untuk integrasi dan akses data jarak jauh mungkin tidak dipertimbangkan secara mendalam.

## Tujuan

Tujuan utama dari pembuatan sistem absensi berbasis fingerprint untuk perhitungan gaji berdasarkan absensi adalah untuk mengembangkan sebuah sistem otomatis yang dapat menghitung gaji berdasarkan absensi dan merekam data absensi yang tepat dan akurat.

## Manfaat

Dengan menggunakan sistem ini, proses absensi dan perhitungan gaji dapat dilakukan secara otomatis dan efisien. Hal ini dapat mengurangi ketergantungan pada pemisahan secara manual yang membutuhkan waktu dan tenaga kerja manusia yang banyak. Dengan demikian, produktivitas dan efisiensi dapat meningkat secara signifikan.

# BAB II

**ANALISIS DAN PERANCANGAN SISTEM**

## Deskripsi Sistem

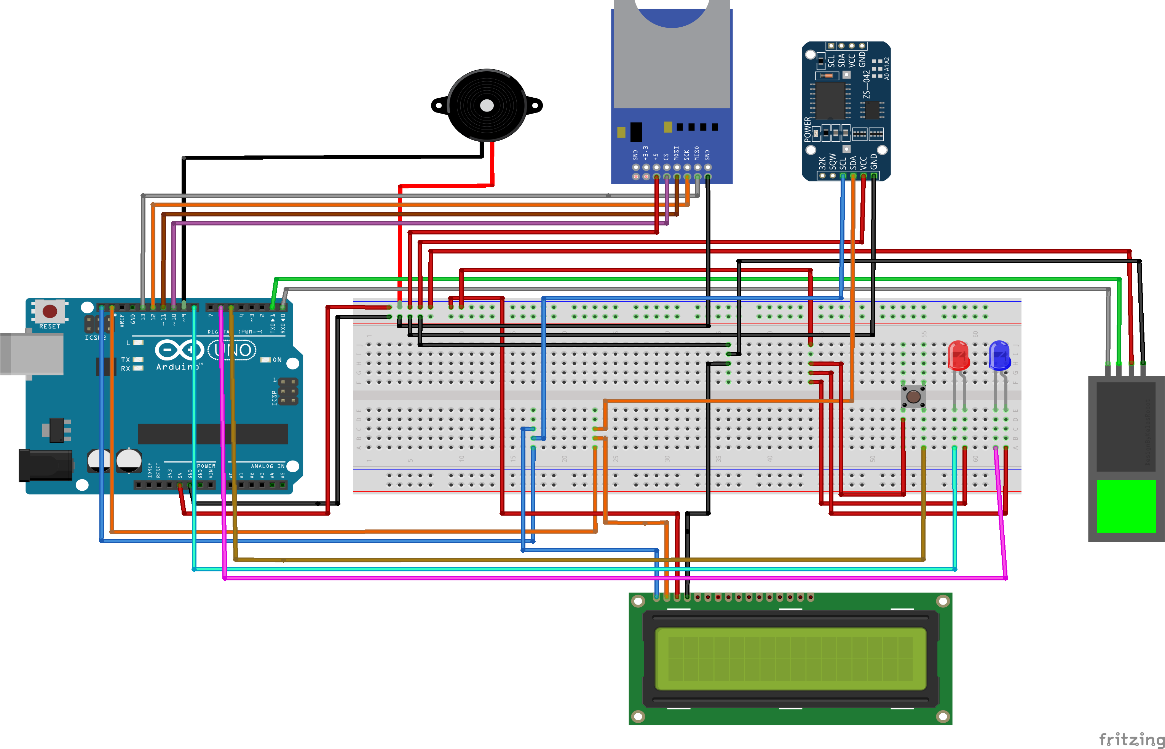
Sistem Absensi berbasis fingerprint untuk perhitungan gaji berdasarkan absensi adalah sebuah perangkat yang menggunakan Arduino sebagai kontrol utama untuk melakukan absensi dan perhitungan gaji berdasarkan absensi nya. Sistem Abensi untuk perhitungan gaji ini biasanya digunakan dalam industri atau proyek-proyek *DIY* untuk melakukan proses absensi dan perhitungan gaji secara otomatis dan efisien. Dengan menggunakan Arduino, pengguna dapat dengan mudah mengembangkan sistem penyortiran berdasarkan warna yang efisien dan akurat.

## Alat dan Bahan

* + - 1. Arduino Uno
      2. Sensor Fingerprint AS608
      3. RTC (Real Time Clock)
      4. LCD 16 x 2
      5. SD Card Module
      6. Breadboard
      7. Kabel Jumper
      8. Lampu LED (merah, biru)
      9. Buzzer
      10. Push Button

7

## Rangkaian



Gambar 1 Rangkaian Fritzing

## Cara Kerja Sistem

1. Untuk menambah sidik jari baru dengan mengklik push button. Kemudian LCD menampilkan silahkan masukan sidik jari, setelah sidik jari dimasukan, maka sensor fingerprint akan merekam sidik jari yang di masukan. Jika tersimpan maka LCD menampilkan data tersimpan jika tidak maka LCD menampilkan tidak tersimpan.
2. Sebelum melakukan proses absensi, LCD menampilkan waktu dan tanggal sesuai dengan waktu indonesia barat yang di ambil dari RTC yang sebelum nya sudah di tentukan.
3. Untuk proses absensi nya, Kami menetapkan waktu masuk dan pulang, untuk masuk sendiri ialah pukul 07.00 hingga 15.00 dan untuk pulang ialah pukul 15.00 hingga 17.00. Kemudian untuk waktu absensi masuk kami

membatasinya, jika absensi lebih dari jam 8 maka keterangan absensi nya

terlambat, lalu jika kurang dari jam 8 maka absensi tepat waktu .

1. Kemudian untuk absensi nya dengan menempelkan sidik jari terhadap sensor fingerprint, setelah menempelkan sidik jari nya kepada sensor fingerprint, maka sensor fingerprint akan mengambil gambar sidik jari lalu sensor fingerprint akan memvalidasi apakah sidik jari tersebut sudah terdaftar. Jika tidak terdaftar maka LCD menampilkan tidak terdaftar, LED merah menyala dan buzzer berbunyi, kemudian jika terdaftar maka akan menampilkan status absensi sesuai dengan waktu absensi dan nama sesuai dengan ID yang tersimpan pada sensor fingerprint dan lampu biru menyala.
2. Lalu untuk proses penyimpanan absensi nya, kami menyimpan semua data baik itu sidik jari terdaftar maupun tidak terdaftar dengan Nama berdasarkan ID, waktu saat absensi, kemudian keterangan absensi.
3. Kemudian untuk proses perhitungan gaji nya, kami menetapkan waktu pada tanggal 1 untuk pehitungan gaji nya, perhitungan gaji ini di hitung jika absensi dengan keterangan absensi tepat waktu, jika terlambat maka tidak di hitung

## Dasar Teori

## Arduino Uno



Gambar 2 Arduino Uno

Arduino Uno adalah sebuah board mikrokontroler yang berbasis Atmega328, memiliki 14 pin input/output yang mana 6 pin dapat digunakan sebagai output PWM, 6 analog input, crystal osilator 16 MHz, port USB, power jack, ICSP header, dan sebuah tombol reset.

## Sensor Fingerprint AS608



Gambar 3 Sensor Fingerprint AS608

Sensor fingerprint AS608 adalah modul pemindai sidik jari yang dirancang untuk mengidentifikasi dan memverifikasi sidik jari dengan akurasi tinggi. Sensor ini tersusun atas sensor optik dan prosesor on-board yang mampu menangkap dan memproses gambar sidik jari secara efisien. Keluaran dari sensor ini adalah data digital yang menunjukkan hasil pencocokan sidik jari dengan template yang tersimpan dalam memori internal sensor. AS608 bekerja dengan cara mengambil gambar sidik jari saat jari ditempatkan pada sensor, kemudian memproses dan membandingkan gambar tersebut dengan data sidik jari yang sudah tersimpan untuk melakukan verifikasi atau identifikasi.

## RTC (Real Time Clock)



Gambar 4 RTC (Real Time Clock)

RTC (Real Time Clock) DS3231 adalah modul jam waktu nyata yang dirancang untuk memberikan informasi waktu yang akurat. Modul ini dilengkapi dengan osilator kristal terintegrasi dan kompensasi suhu yang memastikan presisi tinggi. Keluaran dari RTC DS3231 adalah data waktu dalam format digital yang mencakup detik, menit, jam, hari, tanggal, bulan, dan tahun. RTC ini juga memiliki memori internal untuk menyimpan data waktu saat daya mati dan baterai cadangan untuk menjaga fungsi jam berjalan saat tidak ada daya eksternal.

## LCD 16x2



Gambar 5 LCD 16 x 2

LCD 16x2 adalah modul tampilan yang dapat menampilkan 16 karakter dalam 2 baris, sering digunakan dalam proyek mikrokontroler untuk menampilkan data secara real-time. Modul ini terdiri dari matriks karakter 5x8 titik, yang memungkinkan representasi jelas dari teks dan simbol. Keluaran dari LCD 16x2 adalah tampilan visual dari teks atau data yang dikirimkan dari mikrokontroler melalui antarmuka yang mudah digunakan, biasanya dalam mode 4-bit atau 8-bit.

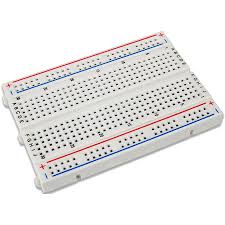
## SD Card Module



Gambar 6 SD Card Module

Modul SD Card adalah komponen yang memungkinkan mikrokontroler untuk membaca dan menulis data ke kartu SD. Modul ini sangat berguna untuk aplikasi yang memerlukan penyimpanan data berkapasitas besar, seperti pencatatan data sensor, penyimpanan file log, dan aplikasi multimedia. Keluaran dari modul SD Card adalah data digital yang disimpan pada kartu SD, yang dapat diakses dan dikelola oleh mikrokontroler melalui antarmuka SPI (Serial Peripheral Interface).

## Breadboard



Gambar 7 Breadboard

Breadboard merupakan sebuah board atau papan yang berfungsi untuk merancang sebuah rangkaian elektronik sederhana. Breadboard tersebut nantinya akan dilakukan prototipe atau uji coba tanpa harus melakukan solder. Salah satu keuntungan menggunakan breadboard adalah komponenkomponen yang dirakit tersebut tidak akan mengalami kerusakan. Komponen tersebut juga masih bisa dirangkai kembali untuk membentuk rangkaian yang lainnya.

## Kabel Jumper



Gambar 8 Kabel Jumper

Kabel jumper adalah kabel kecil yang digunakan untuk membuat koneksi listrik sementara antara komponen-komponen elektronik di papan sirkuit, breadboard, atau perangkat prototyping lainnya. Kabel jumper terdiri dari konduktor logam yang dilapisi dengan isolasi plastik, dan ujung-ujungnya biasanya memiliki pin atau konektor yang memungkinkan mereka untuk dengan mudah disisipkan dan dilepaskan.

## Lampu LED



Gambar 9 Lampu LED

Lampu LED (Light Emitting Diode) adalah komponen elektronik yang mengeluarkan cahaya saat arus listrik mengalir melaluinya. LED sangat efisien dalam hal konsumsi energi dan memiliki umur panjang, menjadikannya pilihan populer untuk berbagai aplikasi pencahayaan dan indikasi dalam proyek-proyek elektronik.

## Buzzer



Gambar 10 Buzzer

## Buzzer adalah perangkat elektronik yang digunakan untuk menghasilkan suara atau sinyal akustik. Buzzer sering digunakan dalam berbagai aplikasi untuk memberikan peringatan, alarm, indikasi status, atau umpan balik auditori dalam perangkat elektronik dan sistem kontrol.

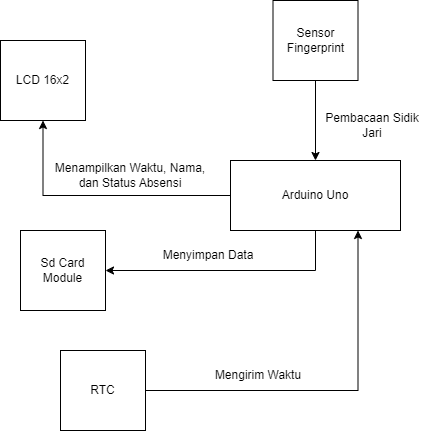
## Push Button



Gambar 11 Push Button

Push button adalah komponen elektronik yang digunakan sebagai saklar mekanis untuk mengendalikan aliran arus listrik dalam suatu rangkaian. Push button bekerja dengan cara menutup sirkuit saat ditekan dan membuka sirkuit saat dilepaskan. Dalam proyek mikrokontroler, push button sering digunakan untuk menerima input dari pengguna, seperti memulai atau menghentikan proses tertentu.

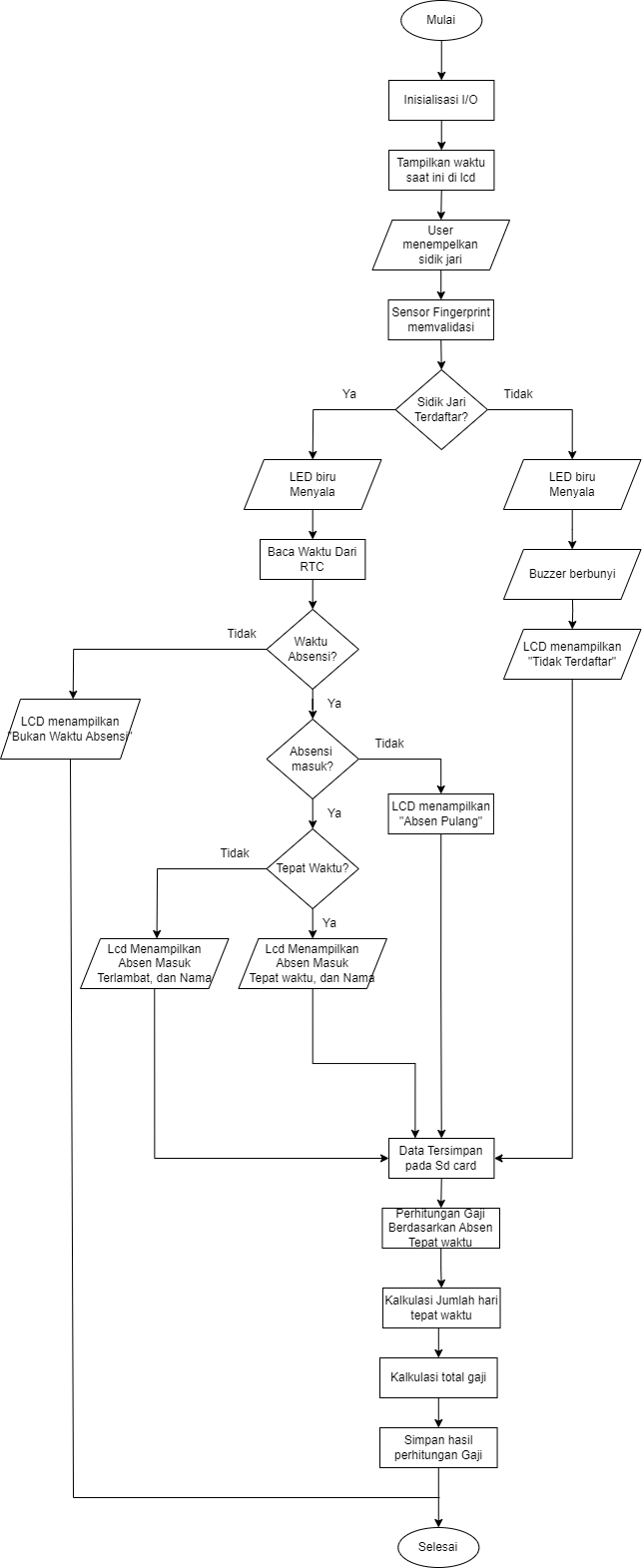
## Diagram Blok Sistem



Gambar 12 Diagram Blok Sistem

Diagram blok di atas adalah bentuk alur perancangan sistem absensi berbasis fingerprint untuk perhitungan gaji berdasarkan absensi. Seperti yang tertera pada gambar di atas, pertama-tama sidik jari karyawan akan dibaca oleh sensor fingerprint AS608. Kemudian, Arduino Uno akan memproses data sidik jari tersebut dan memverifikasi apakah sidik jari tersebut sudah terdaftar atau tidak. Setelah verifikasi, Arduino akan mengambil data waktu dari RTC (Real-Time Clock) dan mencatat waktu absensi. Informasi mengenai waktu, nama, dan status absensi karyawan kemudian akan ditampilkan pada LCD 16x2. Selain itu, data absensi juga akan disimpan pada modul SD Card untuk pencatatan dan perhitungan gaji lebih lanjut. Dengan sistem ini, proses absensi dapat dilakukan secara otomatis, akurat, dan efisien.

## Flowchart



Gambar 13 Flowchart

# BAB III

**HASIL DAN PEMBAHASAN**

## Hasil

Setelah semua komponen dirakit sesuai dengan rangkaian, kemudian dilakukan pengujian kinerja sensor Fingerprint AS608. Langkah pertama adalah menguji sensor fingerprint untuk memastikan bahwa sidik jari dapat direkam dan diidentifikasi dengan akurat. Berikut adalah hasil pengujian dari sistem yang dirakit sebagai berikut:

|  |  |  |
| --- | --- | --- |
| Karyawan | Sidik Jari Terdaftar | Sidik jari tidak terdaftar |
| A | Ya | Tidak |
| B | Ya | Tidak |
| C | Ya | Tidak |
| D | Tidak | Ya |

Gambar 14 Tabel Fingerprint

Dari hasil pengujian di atas, dapat dilihat bahwa sensor Fingerprint AS608 mampu merekam dan mengidentifikasi sidik jari dengan baik. Karyawan A, B, dan C berhasil mendaftarkan sidik jarinya, sedangkan karyawan D tidak terdaftar. Hal ini menunjukkan bahwa sensor fingerprint memiliki kemampuan akurasi yang tinggi dalam membedakan sidik jari yang terdaftar dan tidak terdaftar, sehingga dapat diandalkan untuk keperluan absensi dan identifikasi karyawan.

Kemudian kita coba pengujian Sd Card moodule dilakukan untuk memastikan bahwa dapat absensi dapat disimpan dengan baik. Pengujian dilakukan dengan menyimpan data absensi setiap kali karyawan melakukan absensi. Berikut

18

Adalah contoh data yang disimpan :

|  |  |  |
| --- | --- | --- |
| Karyawan | Waktu Absensi | Status Absensi |
| A | 07:15 | Tepat Waktu |
| B | 07:45 | Terlambat |
| C | 08:00 | Terlambat |
| D | 12:30 | Tidak Terdaftar |

Gambar 15 Tabel Absensi

Data yang tersimpan menunjukkan bahwa SD Card Module mampu menyimpan data absensi dengan baik dan dapat diakses kembali untuk keperluan perhitungan gaji.

Setelah data absensi karyawan tersimpan dalam SD Card Module, Langkah berikut nya adalah melakukan perhitungan gaji berdasarkan data absensi yang telah tercatat. Gaji karyawan dihitung berdasarkan jumlah absen dengan status Tepat Waktu. Jika karyawan terlambat atau tidak hadir, hari tersebut tidak di hitung dalam perhitungan gaji. Berikut sebagai contoh perhitungan gaji :

|  |  |  |
| --- | --- | --- |
| Karyawan | Hari Tepat Waktu | Total Gaji |
| A | 1 | 100.000 |
| B | 0 | 0 |
| C | 0 | 0 |

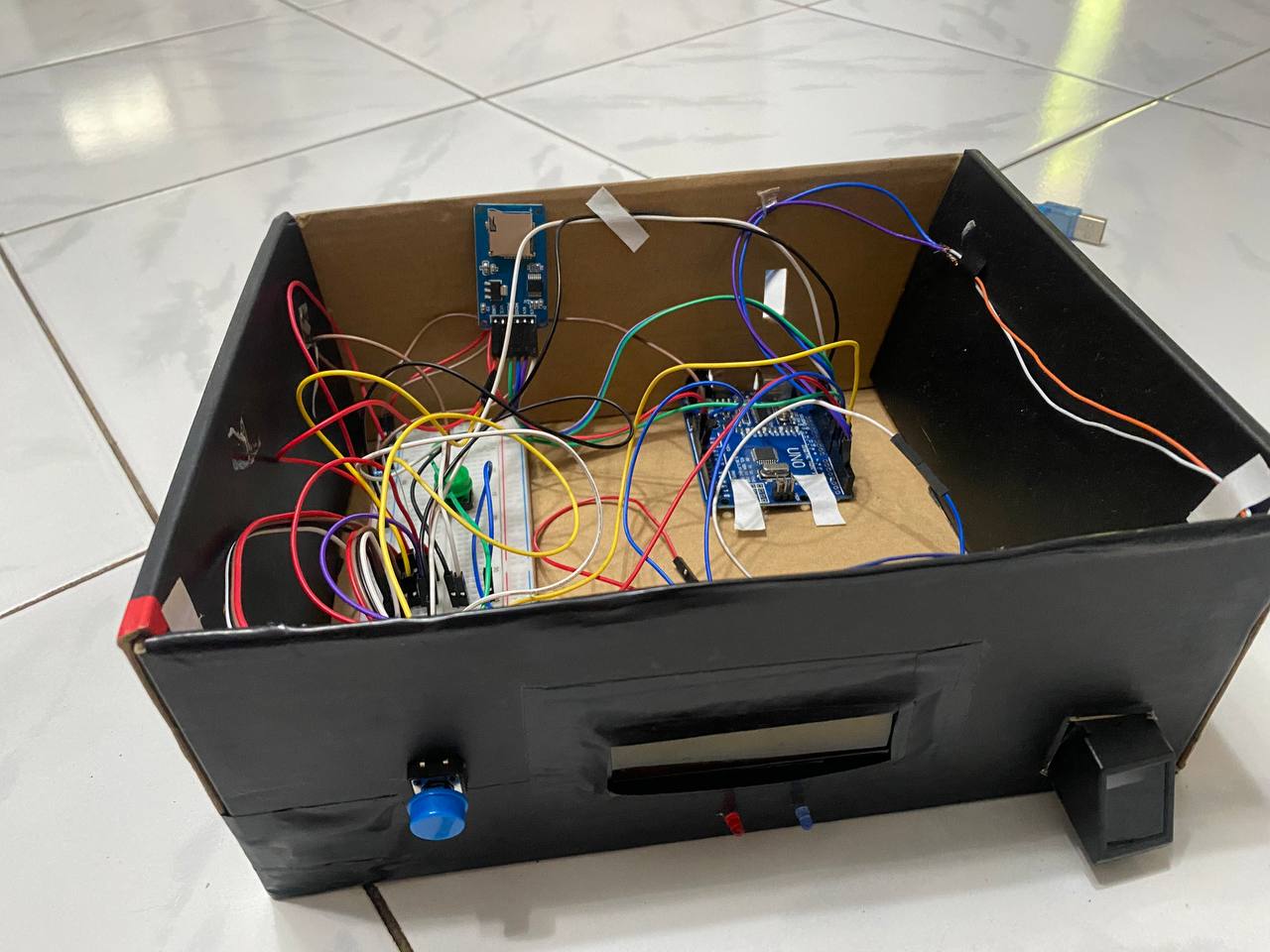
Gambar 16 Tabel Perhitungan Gaji

Data di atas menunjukkan bahwa total gaji dihitung berdasarkan jumlah hari karyawan hadir tepat waktu. Jika karyawan terlambat atau tidak hadir, hari tersebut tidak dihitung dalam perhitungan gaji. Dengan demikian, sistem absensi berbasis fingerprint ini tidak hanya mencatat kehadiran, tetapi juga membantu dalam

perhitungan gaji karyawan secara otomatis dan akurat.

Sistem absensi berbasis fingerprint untuk perhitungan gaji berdasarkan absensi menggunakan Arduino Uno terbukti efektif dan efisien dalam mencatat kehadiran karyawan dan menghitung gaji mereka. Sistem ini dapat meningkatkan akurasi pencatatan kehadiran, mengurangi kesalahan manusia, dan mencegah kecurangan. Implementasi sistem ini dapat membantu perusahaan dalam mengelola kehadiran karyawan dan perhitungan gaji dengan lebih baik.

Berikut adalah hasil perakitan dari Sistem Absensi Berbasis fingerprint untuk perhitungan gaji berdasarkan absensi :



Gambar 17 Hasil Perakitan

## Program

#include <Wire.h> // Library untuk komunikasi I2C

#include <LiquidCrystal\_I2C.h> // Library untuk LCD yang menggunakan I2C

#include <RTClib.h> // Library untuk RTC (Real-Time Clock)

#include <Adafruit\_Fingerprint.h> // Library untuk sensor sidik jari dari Adafruit

#include <SD.h> // Library untuk membaca dan menulis ke kartu SD

#include <SPI.h> // Library untuk komunikasi SPI

#include <avr/pgmspace.h> // Library untuk menyimpan data di memori flash (PROGMEM)

#if (defined(\_\_AVR\_\_) || defined(ESP8266)) && !defined(\_\_AVR\_ATmega2560\_\_)

#include <SoftwareSerial.h> // Library untuk membuat port serial pada pin lain

SoftwareSerial mySerial(2, 3); // Membuat port serial pada pin 2 (RX) dan 3 (TX) jika menggunakan board selain ATmega2560

#else

#define mySerial Serial1 // Mendefinisikan mySerial sebagai Serial1 jika menggunakan ATmega2560

#endif

const int cs = 10; // Mendefinisikan pin chip select untuk kartu SD pada pin 7

const int ledRed = 6; // Mendefinisikan pin untuk LED merah pada pin 6

const int ledBlue = 8; // Mendefinisikan pin untuk LED Biru pada pin 8

const int buzzer = 9; // Mendefinisikan pin untuk buzzer pada pin 8

const int button = 5; // Mendefinisikan pin untuk button pada pin 8

LiquidCrystal\_I2C lcd(0x27, 16, 2); // Menginisialisasi LCD I2C dengan alamat 0x27, ukuran 16x2 karakter

RTC\_DS3231 rtc; // Membuat objek RTC dari kelas RTC\_DS3231

Adafruit\_Fingerprint finger = Adafruit\_Fingerprint(&mySerial); // Membuat objek sensor sidik jari menggunakan port serial yang sudah didefinisikan

bool enrollMode = false; // Deklarasi variabel boolean untuk menentukan apakah mode pendaftaran sidik jari aktif atau tidak. Default-nya adalah false.

const char namaPengguna0[] PROGMEM = "Anida Aulia Utami"; // Menyimpan nama pengguna di memori flash (PROGMEM)

const char namaPengguna1[] PROGMEM = "Rifky Hilman"; // Menyimpan nama pengguna di memori flash (PROGMEM) untuk menghemat SRAM

const char namaPengguna2[] PROGMEM = "Afif Naufal"; // Menyimpan nama pengguna di memori flash (PROGMEM)

const char namaPengguna3[] PROGMEM = "Merliyana"; // Menyimpan nama pengguna di memori flash (PROGMEM)

// Array pointer ke string yang juga disimpan di memori flash

const char\* const namaPengguna[] PROGMEM = {

namaPengguna0,

namaPengguna1,

namaPengguna2,

namaPengguna3,

};

// Fungsi untuk mencetak nama pengguna berdasarkan ID

void printName(int id) {

char buffer[20]; // Mendeklarasikan Array String dengan panjang 20 karakter

strcpy\_P(buffer, (char\*)pgm\_read\_word(&(namaPengguna[id]))); // Membaca string dari memori flash dan menyalinnya ke buffer

lcd.print(buffer); // Mencetak nama pengguna di LCD

}

void setup() {

Wire.begin(); // Memulai komunikasi I2C

Serial.begin(9600); // Memulai komunikasi serial dengan baud rate 9600

lcd.init(); // Inisialisasi LCD

lcd.backlight(); // Menyalakan lampu latar LCD

pinMode(ledRed, OUTPUT); // Menentukan pin ledRed sebagai output

pinMode(ledBlue, OUTPUT); // Menentukan pin ledBlue sebagai output

pinMode(buzzer, OUTPUT); // Menentukan pin buzzer sebagai output

pinMode(button, INPUT\_PULLUP); // Mengatur push button sebagai input dengan pull-up resistor internal

if (!rtc.begin()) { // Memeriksa apakah RTC terhubung

Serial.println(F("RTC tidak ditemukan!"));

lcd.clear(); // Membersihkan layar LCD

lcd.print(F("RTC Error")); // Menampilkan pesan kesalahan pada LCD

while (1); // Berhenti di sini jika RTC tidak ditemukan

}

if (rtc.lostPower()) { // Memeriksa apakah RTC kehilangan daya dan perlu diatur ulang

Serial.println(F("RTC kehilangan daya, set waktu ke default!"));

rtc.adjust(DateTime(F(\_\_DATE\_\_), F(\_\_TIME\_\_))); // Mengatur waktu RTC ke waktu kompilasi sketsa

}

delay(100); // Memberikan jeda waktu 100 ms

Serial.println(F("\n\nAdafruit finger detect test")); // Menampilkan pesan pada serial monitor

finger.begin(57600); // Memulai komunikasi dengan sensor sidik jari pada baud rate 57600

delay(5); // Memberikan jeda waktu 5 ms

if (finger.verifyPassword()) { // Memeriksa apakah sensor sidik jari terhubung dan berfungsi

Serial.println(F("Found fingerprint sensor!"));

} else {

Serial.println(F("Did not find fingerprint sensor :("));

while (1) { delay(1); } // Berhenti di sini jika sensor sidik jari tidak ditemukan

}

Serial.println(F("Reading sensor parameters"));

finger.getParameters(); // Membaca parameter sensor sidik jari

finger.getTemplateCount(); // Mendapatkan jumlah template (sidik jari) yang tersimpan di sensor

if (finger.templateCount == 0) { // Memeriksa apakah ada template yang tersimpan

Serial.println(F("Sensor doesn't contain any fingerprint data. Please run the 'enroll' example."));

} else {

Serial.println(F("Waiting for valid finger..."));

Serial.print(F("Sensor contains ")); Serial.print(finger.templateCount); Serial.println(F(" templates"));

}

if (!SD.begin(cs)) { // Memulai komunikasi dengan kartu SD

Serial.println(F("Inisialisasi SD card gagal!"));

return; // Keluar dari fungsi setup() jika inisialisasi kartu SD gagal

}

Serial.println(F("SD card berhasil diinisialisasi."));

DateTime now = rtc.now();

if (now.day() == 1) { // Jika hari ini adalah tanggal 30

hitungGaji(); // Panggil fungsi hitungGaji untuk menghitung gaji

}

}

void loop() {

DateTime now = rtc.now(); // Mendapatkan waktu dan tanggal saat ini dari RTC (Real-Time Clock)

lcd.setCursor(0, 0); // Mengatur kursor LCD pada baris pertama, kolom pertama

lcd.print(F("Time: ")); // Menampilkan teks "Time: " di LCD

if (now.hour() < 10) lcd.print('0'); // Menambahkan '0' jika jam kurang dari 10 (untuk format HH)

lcd.print(now.hour()); // Menampilkan jam saat ini

lcd.print(':'); // Menampilkan karakter ':'

if (now.minute() < 10) lcd.print('0'); // Menambahkan '0' jika menit kurang dari 10 (untuk format MM)

lcd.print(now.minute()); // Menampilkan menit saat ini

lcd.print(':'); // Menampilkan karakter ':'

if (now.second() < 10) lcd.print('0'); // Menambahkan '0' jika detik kurang dari 10 (untuk format SS)

lcd.print(now.second()); // Menampilkan detik saat ini

lcd.setCursor(0, 1); // Mengatur kursor LCD pada baris kedua, kolom pertama

lcd.print(F("Date: ")); // Menampilkan teks "Date: " di LCD

if (now.day() < 10) lcd.print('0'); // Menambahkan '0' jika hari kurang dari 10 (untuk format DD)

lcd.print(now.day()); // Menampilkan hari saat ini

lcd.print('/'); // Menampilkan karakter '/'

if (now.month() < 10) lcd.print('0'); // Menambahkan '0' jika bulan kurang dari 10 (untuk format MM)

lcd.print(now.month()); // Menampilkan bulan saat ini

lcd.print('/'); // Menampilkan karakter '/'

lcd.print(now.year()); // Menampilkan tahun saat ini

delay(1000); // Memberikan jeda waktu 1 detik sebelum melanjutkan loop

if (enrollMode) { // Memeriksa apakah mode enroll (pendaftaran sidik jari) aktif

enrollFingerprint(); // Jika ya, panggil fungsi enrollFingerprint()

} else {

getFingerprintID(now); // Jika tidak, panggil fungsi getFingerprintID() dengan parameter waktu saat ini

}

if (digitalRead(button) == LOW) { // Jika push button ditekan

enrollMode = true;

}

}

void writeAbsensiToSD(int id, const char\* status, DateTime now) { //function menyimpan absensi terdaftar

File dataFile = SD.open("absensi.csv", FILE\_WRITE); // Membuka file "absensi.txt" dalam mode tulis

if (dataFile) { // Memeriksa apakah file berhasil dibuka

dataFile.print(F("ID: ")); // Menulis teks "ID: " ke file

dataFile.print(id); // Menulis ID pengguna ke file

dataFile.print(F(", Nama: ")); // Menulis teks ", Nama: " ke file

char buffer[20]; // Membuat buffer untuk menyimpan nama pengguna

strcpy\_P(buffer, (char\*)pgm\_read\_word(&(namaPengguna[id]))); // Menyalin nama pengguna dari memori PROGMEM ke buffer

dataFile.print(buffer); // Menulis nama pengguna ke file

dataFile.print(F(", Waktu: ")); // Menulis teks ", Waktu: " ke file

if (now.hour() < 10) dataFile.print('0'); // Menambahkan '0' jika jam kurang dari 10

dataFile.print(now.hour()); // Menulis jam saat ini ke file

dataFile.print(':'); // Menulis karakter ':'

if (now.minute() < 10) dataFile.print('0'); // Menambahkan '0' jika menit kurang dari 10

dataFile.print(now.minute()); // Menulis menit saat ini ke file

dataFile.print(':'); // Menulis karakter ':'

if (now.second() < 10) dataFile.print('0'); // Menambahkan '0' jika detik kurang dari 10

dataFile.print(now.second()); // Menulis detik saat ini ke file

dataFile.print(F(", Tanggal: ")); // Menulis teks ", Tanggal: " ke file

if (now.day() < 10) dataFile.print('0'); // Menambahkan '0' jika hari kurang dari 10

dataFile.print(now.day()); // Menulis hari saat ini ke file

dataFile.print('/'); // Menulis karakter '/'

if (now.month() < 10) dataFile.print('0'); // Menambahkan '0' jika bulan kurang dari 10

dataFile.print(now.month()); // Menulis bulan saat ini ke file

dataFile.print('/'); // Menulis karakter '/'

dataFile.print(now.year()); // Menulis tahun saat ini ke file

dataFile.print(F(", Absensi: ")); // Menulis teks ", Absensi: " ke file

dataFile.print(status); // Menulis status absensi ke file

dataFile.println(); // Menulis newline ke file

dataFile.close(); // Menutup file

} else {

Serial.println(F("Gagal membuka file absensi.txt")); // Jika file tidak bisa dibuka, cetak pesan error

}

}

void writeAbsensiNotFound(const char\* nama, DateTime now) { //Function menyimpan absensi yang tidak terdaftar

// Membuka file "notfound.csv" dalam mode tulis. Jika file tidak ada, maka akan dibuat.

File dataFile = SD.open("notfound.csv", FILE\_WRITE);

if (dataFile) { // Memeriksa apakah file berhasil dibuka

dataFile.print(F("Nama: ")); // Menulis teks "Nama: " ke file

dataFile.print(nama); // Menulis nama yang diberikan ke file

dataFile.print(F(", Waktu: ")); // Menulis teks ", Waktu: " ke file

// Menulis jam dengan format HH:MM:SS

if (now.hour() < 10) dataFile.print('0'); // Menambahkan '0' jika jam kurang dari 10

dataFile.print(now.hour()); // Menulis jam saat ini ke file

dataFile.print(':'); // Menulis karakter ':'

if (now.minute() < 10) dataFile.print('0'); // Menambahkan '0' jika menit kurang dari 10

dataFile.print(now.minute()); // Menulis menit saat ini ke file

dataFile.print(':'); // Menulis karakter ':'

if (now.second() < 10) dataFile.print('0'); // Menambahkan '0' jika detik kurang dari 10

dataFile.print(now.second()); // Menulis detik saat ini ke file

dataFile.print(F(", Tanggal: ")); // Menulis teks ", Tanggal: " ke file

// Menulis tanggal dengan format DD/MM/YYYY

if (now.day() < 10) dataFile.print('0'); // Menambahkan '0' jika hari kurang dari 10

dataFile.print(now.day()); // Menulis hari saat ini ke file

dataFile.print('/'); // Menulis karakter '/'

if (now.month() < 10) dataFile.print('0'); // Menambahkan '0' jika bulan kurang dari 10

dataFile.print(now.month()); // Menulis bulan saat ini ke file

dataFile.print('/'); // Menulis karakter '/'

dataFile.print(now.year()); // Menulis tahun saat ini ke file

dataFile.println(); // Menulis newline ke file

dataFile.close(); // Menutup file

} else {

// Jika file tidak bisa dibuka, cetak pesan error ke Serial Monitor

Serial.println(F("Gagal membuka file AbsenTidakTerdaftar.txt"));

}

}

void hitungGaji() {

int totalGaji[sizeof(namaPengguna) / sizeof(namaPengguna[0])] = {0}; // Array untuk menyimpan total gaji

bool dataDiperbarui[sizeof(namaPengguna) / sizeof(namaPengguna[0])] = {false}; // Array untuk menandai data yang sudah diperbarui

File absensiFile = SD.open("absensi.csv", FILE\_READ); // Membuka file "absensi.txt" dalam mode baca

if (absensiFile) { // Memeriksa apakah file berhasil dibuka

while (absensiFile.available()) { // Membaca file sampai selesai

String line = absensiFile.readStringUntil('\n'); // Membaca satu baris dari file

if (line.startsWith("ID: ")) { // Jika baris dimulai dengan "ID: "

int id = line.substring(4, line.indexOf(',')).toInt(); // Mendapatkan ID dari baris

String status = line.substring(line.indexOf("Absensi: ") + 9); // Mendapatkan status absensi dari baris

if (status.startsWith("Absen Masuk Tepat")) { // Jika status adalah "Absen Masuk Tepat"

totalGaji[id] += 1; // Menambah jumlah masuk tepat waktu

}

}

}

absensiFile.close(); // Menutup file absensi

// Membaca dan mengupdate data gaji di file gaji.txt

File gajiFile = SD.open("gaji.csv", FILE\_WRITE); // Membuka file "gaji.txt" dalam mode tulis

if (gajiFile) { // Memeriksa apakah file berhasil dibuka

bool isFirstLine = true; // Flag untuk baris pertama

while (gajiFile.available()) { // Membaca file sampai selesai

String line = gajiFile.readStringUntil('\n'); // Membaca satu baris dari file

if (isFirstLine) { // Jika ini adalah baris pertama

isFirstLine = false;

continue; // Lewatkan baris pertama

}

int id = line.substring(line.indexOf("Nama: ") + 6, line.indexOf(", Jumlah Masuk Tepat Waktu: ")).toInt(); // Mendapatkan ID dari baris

int jumlahMasukTepat = line.substring(line.indexOf("Jumlah Masuk Tepat Waktu: ") + 27, line.indexOf(", Jumlah Gaji: ")).toInt(); // Mendapatkan jumlah masuk tepat waktu dari baris

int jumlahGaji = line.substring(line.indexOf("Jumlah Gaji: ") + 13).toInt(); // Mendapatkan jumlah gaji dari baris

if (id >= 0 && id < sizeof(namaPengguna) / sizeof(namaPengguna[0])) { // Jika ID valid

char buffer[20]; // Membuat buffer untuk menyimpan nama pengguna

strcpy\_P(buffer, (char\*)pgm\_read\_word(&(namaPengguna[id]))); // Menyalin nama pengguna dari memori PROGMEM ke buffer

if (!dataDiperbarui[id]) { // Jika data belum diperbarui

totalGaji[id] += jumlahMasukTepat; // Menghitung ulang jumlah masuk tepat waktu

jumlahGaji = totalGaji[id] \* 100000; // Menghitung ulang total gaji

dataDiperbarui[id] = true; // Menandai data sudah diperbarui

}

gajiFile.print(buffer); // Menulis nama pengguna ke file

gajiFile.print(F(", Jumlah Masuk Tepat Waktu: ")); // Menulis teks ", Jumlah Masuk Tepat Waktu: " ke file

gajiFile.print(totalGaji[id]); // Menulis jumlah masuk tepat waktu ke file

gajiFile.print(F(", Jumlah Gaji: ")); // Menulis teks ", Jumlah Gaji: " ke file

gajiFile.println(jumlahGaji); // Menulis jumlah gaji ke file

}

}

// Menambahkan data baru jika belum ada di file gaji.txt

for (int i = 1; i < sizeof(namaPengguna) / sizeof(namaPengguna[0]); i++) {

if (!dataDiperbarui[i]) { // Jika data belum diperbarui

char buffer[20]; // Membuat buffer untuk menyimpan nama pengguna

strcpy\_P(buffer, (char\*)pgm\_read\_word(&(namaPengguna[i]))); // Menyalin nama pengguna dari memori PROGMEM ke buffer

gajiFile.print(buffer); // Menulis nama pengguna ke file

gajiFile.print(F(", Jumlah Masuk Tepat Waktu: ")); // Menulis teks ", Jumlah Masuk Tepat Waktu: " ke file

gajiFile.print(totalGaji[i]); // Menulis jumlah masuk tepat waktu ke file

gajiFile.print(F(", Jumlah Gaji: ")); // Menulis teks ", Jumlah Gaji: " ke file

gajiFile.println(totalGaji[i] \* 100000); // Menulis jumlah gaji ke file

}

}

gajiFile.close(); // Menutup file gaji

Serial.println(F("Data gaji berhasil diperbarui di gaji.txt")); // Mencetak pesan sukses

} else {

Serial.println(F("Gagal membuka file gaji.txt untuk mengupdate data gaji")); // Jika file tidak bisa dibuka, cetak pesan error

}

} else {

Serial.println(F("Gagal membuka file absensi.txt untuk menghitung gaji")); // Jika file tidak bisa dibuka, cetak pesan error

}

}

uint8\_t getFingerprintID(DateTime now) { //Function untuk mendapatkan sidik jari absensi

uint8\_t p = finger.getImage(); // Mengambil gambar sidik jari

switch (p) {

case FINGERPRINT\_OK:

Serial.println(F("Image taken"));

break;

case FINGERPRINT\_NOFINGER:

Serial.println(F("No finger detected"));

return p;

case FINGERPRINT\_PACKETRECIEVEERR:

Serial.println(F("Communication error"));

return p;

case FINGERPRINT\_IMAGEFAIL:

Serial.println(F("Imaging error"));

return p;

default:

Serial.println(F("Unknown error"));

return p;

}

p = finger.image2Tz(); // Mengonversi gambar ke template sidik jari

switch (p) {

case FINGERPRINT\_OK:

Serial.println(F("Image converted"));

break;

case FINGERPRINT\_IMAGEMESS:

Serial.println(F("Image too messy"));

return p;

case FINGERPRINT\_PACKETRECIEVEERR:

Serial.println(F("Communication error"));

return p;

case FINGERPRINT\_FEATUREFAIL:

Serial.println(F("Could not find fingerprint features"));

return p;

case FINGERPRINT\_INVALIDIMAGE:

Serial.println(F("Could not find fingerprint features"));

return p;

default:

Serial.println(F("Unknown error"));

return p;

}

p = finger.fingerSearch(); // Mencari kecocokan sidik jari di dalam database

if (p == FINGERPRINT\_OK) {

Serial.println(F("Found a print match!"));

lcd.clear();

lcd.setCursor(0, 0);

digitalWrite(ledBlue, HIGH); // Menyalakan LED merah

delay(1000);

digitalWrite(ledBlue, LOW); // Mematikan LED merah

const char\* status;

if (now.hour() >= 7 && now.hour() < 15) { // Jika waktu sekarang antara jam 8 dan 10

if (now.hour() >= 8) { // Jika waktu sekarang lebih besar atau sama dengan jam 8

lcd.print(F("Absen Masuk "));

lcd.setCursor(0, 1);

lcd.print(F("Terlambat "));

delay(1000);

status = "Absen Masuk Terlambat"; // Mengatur status absen terlambat

} else {

lcd.print(F("Absen Masuk "));

lcd.setCursor(0, 1);

lcd.print(F("Tepat Waktu"));

delay(1000);

status = "Absen Masuk Tepat"; // Mengatur status absen tepat

}

} else if (now.hour() >= 15 && now.hour() < 17) { // Jika waktu sekarang antara jam 11 dan 24

lcd.print(F("Absen Pulang"));

status = "Absen Pulang"; // Mengatur status absen pulang

} else {

lcd.print(F("Bukan Waktu Absensi"));

delay(2000);

lcd.clear();

return p; // Mengembalikan nilai p jika bukan waktu absensi

}

lcd.setCursor(0, 1);

printName(finger.fingerID); // Menampilkan nama berdasarkan ID sidik jari

writeAbsensiToSD(finger.fingerID, status, now); // Menulis data absensi ke SD card

delay(2000);

lcd.clear();

} else if (p == FINGERPRINT\_PACKETRECIEVEERR) {

Serial.println(F("Communication error"));

return p; // Mengembalikan nilai p jika terjadi kesalahan komunikasi

} else if (p == FINGERPRINT\_NOTFOUND) {

Serial.println(F("Did not find a match"));

lcd.clear();

digitalWrite(ledRed, HIGH); // Menyalakan LED merah

digitalWrite(buzzer, HIGH);

lcd.print(F("Tidak terdaftar"));

delay(2000);

digitalWrite(buzzer, LOW);

digitalWrite(ledRed, LOW); // Mematikan LED merah

lcd.clear();

writeAbsensiNotFound("Tidak Terdaftar", now);

return p; // Mengembalikan nilai p jika tidak ditemukan kecocokan sidik jari

} else {

Serial.println(F("Unknown error"));

return p; // Mengembalikan nilai p jika terjadi kesalahan tidak dikenal

}

Serial.print(F("Found ID #"));

Serial.print(finger.fingerID);

Serial.print(F(" with confidence of "));

Serial.println(finger.confidence); // Menampilkan ID dan confidence dari kecocokan sidik jari

return finger.fingerID; // Mengembalikan ID sidik jari yang ditemukan

}

void enrollFingerprint() { //function mendaftarkan absensi

int id = finger.templateCount + 1; // Menentukan ID baru untuk sidik jari yang akan didaftarkan

lcd.clear();

lcd.print(F("Masukan Sidik Jari"));

int p = -1;

while (p != FINGERPRINT\_OK) {

p = finger.getImage(); // Mengambil gambar sidik jari

if (p == FINGERPRINT\_NOFINGER) continue; // Jika tidak ada jari, lanjutkan loop

if (p == FINGERPRINT\_IMAGEFAIL) {

lcd.clear();

lcd.print(F("Gagal ambil gambar"));

delay(2000);

return; // Jika gagal mengambil gambar, keluar dari fungsi

}

}

p = finger.image2Tz(1); // Mengonversi gambar ke template sidik jari

if (p != FINGERPRINT\_OK) {

lcd.clear();

lcd.print(F("Gagal konversi"));

delay(2000);

return; // Jika gagal mengonversi gambar, keluar dari fungsi

}

lcd.clear();

lcd.print(F("Lepaskan jari"));

delay(2000);

while (p != FINGERPRINT\_NOFINGER) {

p = finger.getImage(); // Menunggu jari dilepaskan

}

lcd.clear();

lcd.print(F("Masukkan jari"));

delay(2000);

while (p != FINGERPRINT\_OK) {

p = finger.getImage(); // Mengambil gambar sidik jari lagi

if (p == FINGERPRINT\_NOFINGER) continue; // Jika tidak ada jari, lanjutkan loop

if (p == FINGERPRINT\_IMAGEFAIL) {

lcd.clear();

lcd.print(F("Gagal ambil gambar"));

delay(2000);

return; // Jika gagal mengambil gambar, keluar dari fungsi

}

}

p = finger.image2Tz(2); // Mengonversi gambar ke template sidik jari lagi

if (p != FINGERPRINT\_OK) {

lcd.clear();

lcd.print(F("Gagal konversi"));

delay(2000);

return; // Jika gagal mengonversi gambar, keluar dari fungsi

}

p = finger.createModel(); // Membuat model sidik jari

if (p != FINGERPRINT\_OK) {

lcd.clear();

lcd.print(F("Gagal membuat model"));

delay(2000);

return; // Jika gagal membuat model, keluar dari fungsi

}

p = finger.storeModel(id); // Menyimpan model sidik jari ke database

if (p == FINGERPRINT\_OK) {

lcd.clear();

lcd.print(F("tersimpan"));

delay(2000);

lcd.clear();

} else {

lcd.clear();

lcd.print(F("Gagal menyimpan"));

delay(2000);

lcd.clear(); // Jika gagal menyimpan model, tampilkan pesan error

}

enrollMode = false; // Mengatur mode pendaftaran ke false

}