



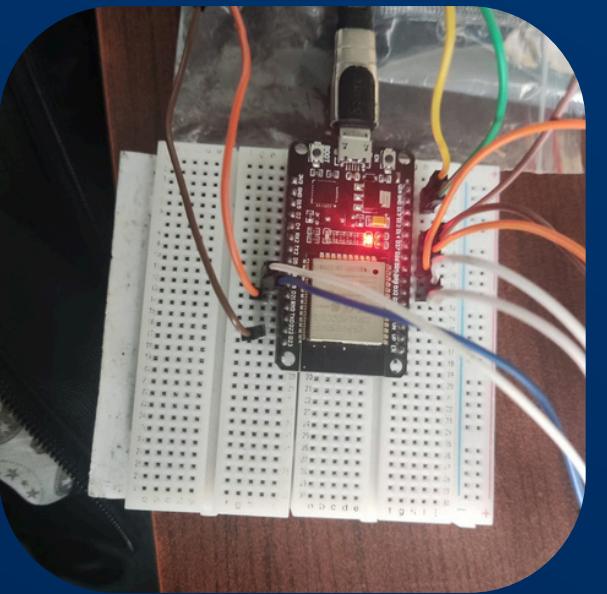
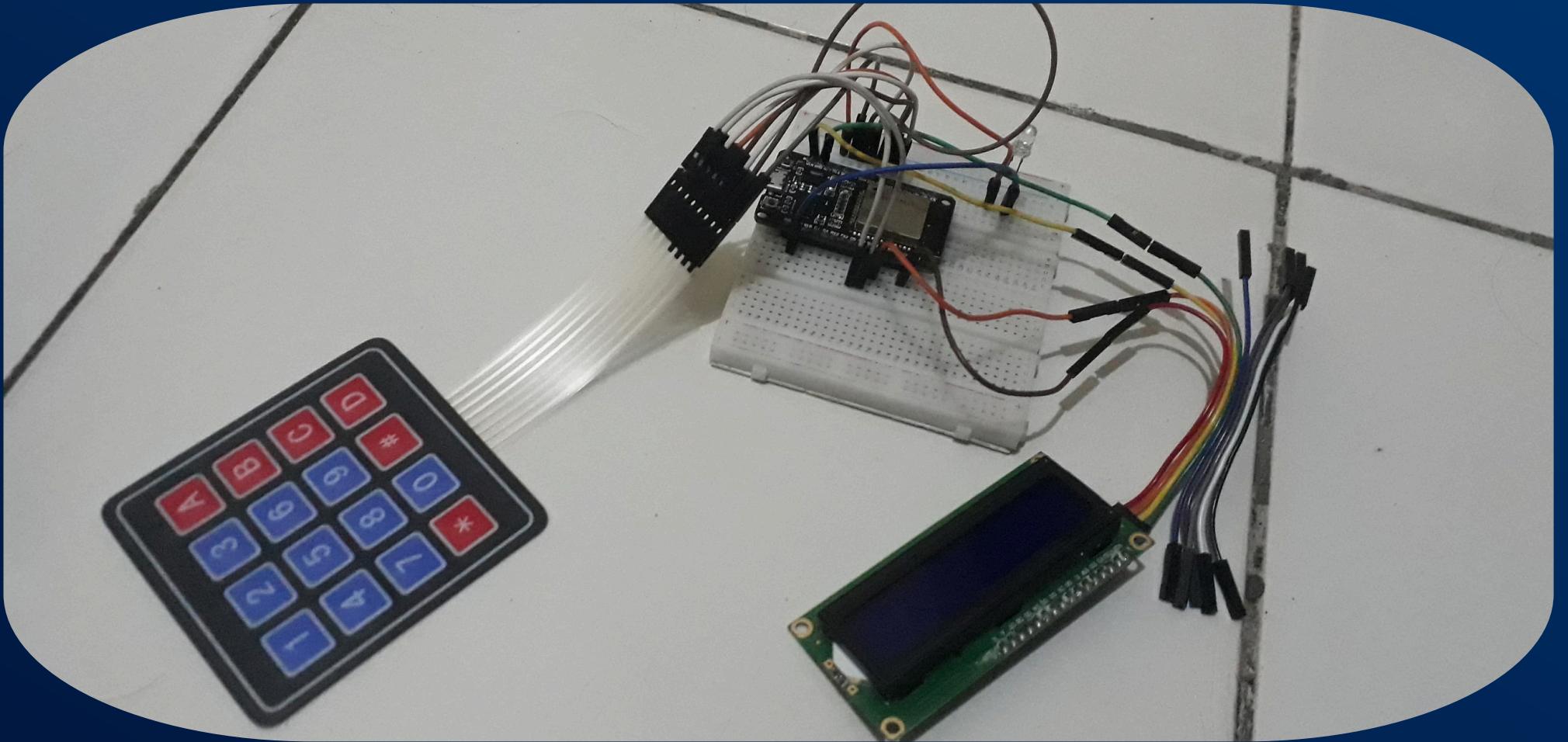
Semester Genap 2023/2024

Proyek SBM 1: Password Manager

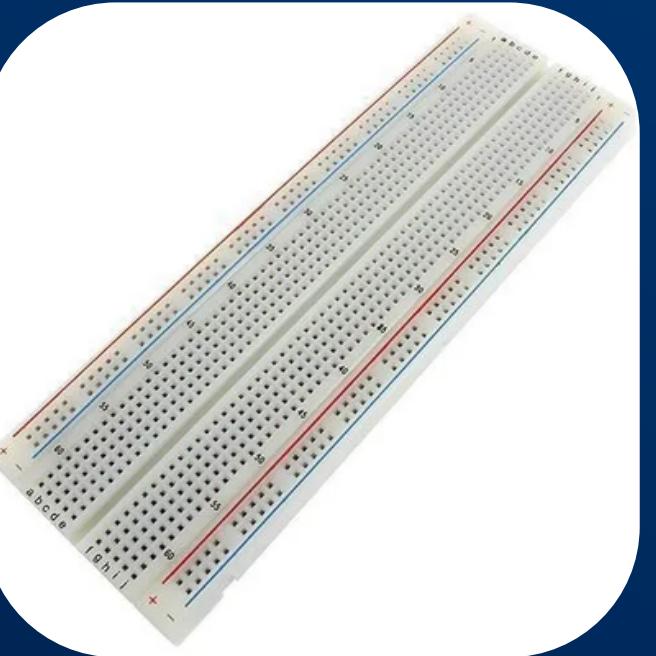
Farhan Arief Ramadhan (22/497297/TK/54496)

Naufal Septio (22/502670/TK/54886)

Rangkaian



Alat & Bahan



Libraries



PasswordManager.hpp

Ada di folder PasswordManager/src

File ini digunakan untuk deklarasi apa saja yang ada di library. Dalam project ini, library PasswordManager berisi sebuah class bernama PasswordManager

```
#ifndef PASSWORDMANAGER_HPP
#include <string>

class PasswordManager{
public:
    static void SetPassword(String pass);
    static bool CheckPassword(String pass);
private:
    static String password;
};

#endif
```

PasswordManager.hpp

PasswordManager.cpp

Ada di folder PasswordManager/src

File ini digunakan untuk deklarasi fungsionalitas
PasswordManager.hpp

```
#include "PasswordManager.hpp"  
  
#include <string>  
  
String PasswordManager::password = "";  
  
void PasswordManager::SetPassword(String pass){  
    PasswordManager::password = pass;  
}  
  
bool PasswordManager::CheckPassword(String pass){  
    return pass == PasswordManager::password;  
}
```

PasswordManager.cpp

Penggunaan Static Library

hpp dan cpp di atas di-build menjadi masing-masing .o (file binary). Kemudian, kedua file .o tersebut di-archive menjadi PasswordManager.a

File .a di-copy ke project utama dan di-configure (di build_flags platformio.ini) agar bisa di-reference menjadi library PasswordManager.hpp (sesuai nama file hpp).

```
[env:esp32dev]
platform = espressif32
board = esp32dev
framework = arduino
lib_deps =
    chris--a/Keypad@^3.1.1
    marcoschwartz/LiquidCrystal_I2C@^1.1.4
build_flags =
    -Ilib/PasswordManager
    -Llib/PasswordManager
    -lPasswordManager
```

platformio.ini di projek utama



Source Code

Library pada Source Code

Disini kami menggunakan 6 library, diantaranya 5 library built-in dan 1 library buatan kami, PasswordManager.hpp.

```
#include <Wire.h>
#include <Arduino.h>
#include <LiquidCrystal_I2C.h>
#include <Keypad.h>
#include <string>
#include <PasswordManager.hpp>
```

```
enum State {  
    Checking,  
    Setting  
};
```

State

Disini kami membuat dua state, yaitu checking dan setting. Checking berfungsi untuk mengecek apakah password benar atau salah, sedangkan setting berfungsi untuk menetapkan password.

Default

Disini terdapat string bernama data dan master.

Data merupakan yang nantinya akan diisi password sedangkan master merupakan password yang tersimpan, disini untuk defaultnya adalah 12345ABC untuk passwordnya

```
std::string data;
std::string master = {"12345ABC"};
byte dataCount = 0, masterCount = 0;
bool pass_is_good;
char customKey;
```

Konfigurasi Keypad

```
const byte ROWS = 4;
const byte COLS = 4;

byte rowPins[ROWS] = {14, 27, 26, 25};
byte colPins[COLS] = {33, 32, 18, 19};

char keys[ROWS][COLS] = {
    '1', '2', '3', 'A',
    '4', '5', '6', 'B',
    '7', '8', '9', 'C',
    '*', '0', '#', 'D'
};

Keypad customKeypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);
```

Disini kami menggunakan keypad 4x4, maka untuk baris dan kolom masing-masing adalah 4.

Untuk baris akan terhubung ke pin 14, 27, 26, dan 25. Sedangkan kolom ke 33, 32, 18, dan 19.

Fungsi clearData berfungsi untuk membersihkan data ketika mengganti state

```
void clearData(){  
    dataCount = 0;  
    data = "";  
    return;  
}
```

Fungsi StoreData berfungsi untuk menyimpan data ketika mengganti password pada state setting

```
void StoreData(char input){  
    dataCount++;  
    data += input;  
}
```

Berfungsi untuk membaca data yang dimasukkan

```
char GetData(){  
    return data[dataCount - 1];  
}
```

Switch State

Fungsi SwitchState berfungsi untuk menginisialisasi state dari dua state yang ada. Pertama data dan tampilan lcd diclear-kan terlebih dahulu,

Setiap state memiliki pesan instruction unik untuk ditampilkan.

Variabel instruction diganti sesuai state lalu di-print pada cursor (0,0).

Selanjutnya untuk state Checking akan menampilkan “Masukkan Sandi:” dan meminta pengguna memasukkan sandi yang telah ditetapkan dan state Setting menampilkan “Sandi Baru:” dan meminta pengguna menetapkan sandi baru.

```
void switchState(State newState){  
    state = newState;  
    clearData();  
    lcd.clear();  
    lcd.setCursor(0,0);  
    String instruction = "~";  
    switch (newState)  
    {  
        case Checking:  
            instruction = "Masukkan Sandi:";  
            break;  
        case Setting:  
            instruction = "Sandi Baru:";  
            break;  
    }  
    lcd.print(instruction);  
}
```

Fungsi Setup

```
int i = 0;

void setup() {
    passwordManager.setPassword(master);

    Serial.begin(9600);
    lcd.init();
    lcd.backlight();
    pinMode(signalPin, OUTPUT);

    switchState(Checking);
}
```

Dipanggil metode SetPassword pada library passwordManager dengan parameter master. Ini berarti kata sandi utama atau master diset disini.

Lalu kita lakukan komunikasi serial dengan baud rate 9600.

Selanjutnya kita inisiasi untuk menampilkan pada layar dengan perintah lcd.init() dan lcd.backlight().

Lalu kita atur mode pin signalPin sebagai output. Ini artinya pin akan digunakan untuk mengirim sinyal dari ESP32.

Fungsi Setup

Pada kondisi default akan menampilkan “Masukkan Sandi:” dan meminta pengguna memasukkan sandi, namun jika menekan tombol “*” akan menuju state setting untuk menetapkan atau mengganti password.

Untuk case dengan state Checking disini pengguna akan diminta memasukkan sandi, setelah itu sandi yang dimasukkan akan dibaca apakah sama dengan master, jika berhasil lcd akan menampilkan “Berhasil” dan dioda akan berkedip, namun jika salah akan menampilkan “Gagal”.

Untuk case dangan state Setting, merupakan kondisi untuk mengubah atau menetapkan password. Pengguna memasukkan password yang diinginkan lalu jika sudah lcd akan menampilkan “Sandi Diubah!” yang artinya pengguna berhasil mengganti sandi dan sandi baru telah tersimpan.

```
void loop() {
    customKey = customKeypad.getKey();
    if (customKey != NO_KEY){

        if (state != Setting && customKey == '*'){
            SwitchState(Setting);
        }

        switch (state)
        {
            case Checking:
                StoreData(customKey);
                lcd.setCursor(dataCount - 1,1);
                lcd.print(GetData());
                if(dataCount == master.length()){
                    lcd.clear();
                    if(passwordManager.CheckPassword(data)){
                        lcd.print("Berhasil");
                        digitalWrite(signalPin, HIGH);
                        delay(2000);
                        digitalWrite(signalPin, LOW);
                    }
                    else{
                        lcd.print("Gagal");
                        delay(1000);
                    }
                    SwitchState(Checking);
                }
                break;
            case Setting:
                if (customKey != '*'){
                    StoreData(customKey);
                    lcd.setCursor(dataCount - 1,1);
                    lcd.print(GetData());
                    Serial.print(dataCount);
                    if(dataCount >= 8){
                        passwordManager.SetPassword(data);
                        Serial.print("Sandi Diubah!");
                        lcd.clear();
                        lcd.print("Sandi Diubah!");
                        delay(1000);
                        SwitchState(Checking);
                    }
                }
                break;
        }
    }
}
```

```
case Checking:  
    StoreData(customKey);  
    lcd.setCursor(dataCount - 1,1);  
    lcd.print(GetData());  
    if(dataCount == master.length()){  
        lcd.clear();  
        if(passwordManager.CheckPassword(data)){  
            lcd.print("Berhasil");  
            digitalWrite(signalPin, HIGH);  
            delay(2000);  
            digitalWrite(signalPin, LOW);  
        }  
        else{  
            lcd.print("Gagal");  
            delay(1000);  
        }  
        SwitchState(Checking);  
    }  
    break;
```

```
case Setting:  
    if (customKey != '*'){  
        StoreData(customKey);  
        lcd.setCursor(dataCount - 1,1);  
        lcd.print(GetData());  
        Serial.print(dataCount);  
        if(dataCount >= 8){  
            passwordManager.SetPassword(data);  
            Serial.print("Sandi Diubah!");  
            lcd.clear();  
            lcd.print("Sandi Diubah!");  
            delay(1000);  
            SwitchState(Checking);  
        }  
    }  
    break;
```

Thank
you very
much!

