

DAFTAR PUSTAKA

- [1] Amir Hamzah.2021.Prototipe Sistem Pembayaran Berbasis RFID Menggunakan Arduino Uno Pada Vending Mechine
- [2] Gunawan Wibisono.2020. Analisa Dan Perancangan Sistem Aplikasi Pembayaran Administrasi Menggunakan Rfid Berbasis Client
- [3] Saputra, D., Andry Iman Juliswanto.2020. Sistem Perhitungan Waktu Rental Studio Musik Berbasis Arduino Menggunakan Validasi RFID,Vol 9 No 1. 0.
- [4] Respati Prayoga,Agung Wibisono.2019. Penghematan Daya Listrik Pada Lampu Dan Sound System Di Studio Musik,prosiding seminar Teknik Elektro Volume 4.
- [5] Tholib Hariono , Hilyah Ashoumi.2020. Implementasi RFID Mobile untuk Alat Pembayaran dan Pengontrol Keuangan Santri, Vol.9 No.2 ISSN : 2338-6313 164
- [6] Mochamad Fajar Wicaksono. 2017. Implementasi Modul Wifi NodeMCU ESP32 untuk Smart Home. Jurnal Teknik Komputer Unikom – Komputika – Volume 6, No.1 – 2017.
- [7] CD Tulle. 2017. Bab II Dasar Teori NodeMCU. Jurnal STMIK Akakom. Vol. 07 No. 01.
- [8] Ahyar Jadid, Zulhelmi, Ardiansyah. 2017. Rancang Bangun Sistem Absensi Perkuliahan Auto ID Berbasis RFID yang Terintegrasi dengan Database Berbasis WEB. Vol.2 No.2 2017: 59-69.
- [9] Hobby Electronics.2020. Jurnal Modul RC522 RFID *Module*. Amikom , Hal.15-20
- [10]Yuga Hadfridar Putra, Dedi Triyanto, Suhard. 2018. Sistem Pemantauan dan Pengendalian Nutrisi, suhu, dan Tinggi air pada Pertanian Hidroponik Berbasis Website. Volume 06, No. 03 (2018), Hal 128-138.

- [11]MR microbot. 2019. *Datasheet* modul Relay. Jurnal STMIK Akakom V01.08 No.01
- [12]Figa Undala, Dedi Triyanto, Yulrio Brianorman. 2019. Prototype Sistem Keamanan Pintu Menggunakan Radio Frequency Identification (RFID) Dengan Kata Sandi Berbasis Mikrokontroler. Jurusan Sistem Komputer, Fakultas MIPA Universitas Tanjungpura
- [13]Faizal Zulmi, 2020. Rancang Bangun Alat Pendeteksi Jarak Aman Pada Kendaraan Berbasis Arduino. Program Studi Teknik Elektro – Fakultas Teknik Universitas Mercu Buana, Jakarta 1,2
- [14]H. S. Weku, E. V. C. Poekoel, R. F. Robot, 2015. “Rancang Bangun Alat Pemberi Pakan Ikan Otomatis Berbasis Mikrokontroler,” J. Tek. Elektro dan Komput., vol. 4, no. 7, pp. 54–64.
- [15]Amin, Fatkhul Nur, 2016. Timbangan Berbasis Arduino dengan Output LCD dan Suara. Diss. Universitas Negeri Semarang,
- [16]Ilman Hidayat, 2014. DFPlayer_Mini_SKU:DFR0299
- [17]H. Hidayat and A. Nugraha, 2015. “Perancangan Perangkat Elektronik Media Pembelajaran Iqra dalam Kode Braille,” J. Sist. Komput., vol. 5, no. 2, pp. 65–71,
- [18]Waluyanti, Sri, dkk. 2008. E-Book Teknik Audio Video. Direktorat Pembinaan SMK.
- [19]Yanto, Mingki., & Joewono.A. 2007.Alat Pengetesan Kurva Polarisasi Speaker. Widya Teknik, 163-172.
- [20]Efrianto, Ridwan, Iman Fahruzi, 2017. Sistem Pengaman Motor Menggunakan Smartcard Politeknik Negeri Batam. Batam Polytechnics Electrical Engineering Study Program
- [21]P.A. Malvino, 2015. Prinsip-prinsip Elektronika. Jakarta: Penerbit Erlangga, 1996

LAMPIRAN A

```
#include <WiFi.h>
#include <HTTPClient.h>
#include <LiquidCrystal_I2C.h>
#include <MFRC522.h>
#include <Keypad_I2C.h>
#include <Keypad.h>
#include <Wire.h>
#include <CountUpDownTimer.h>
#include <EEPROM.h>
#include <SoftwareSerial.h>
#include <DFPlayer_Mini_Mp3.h>
#define HOUR 1
#define MINUTE 2
#define SECOND 3
#define TIMER_SYS HOUR
const byte ROWS = 4;
const byte COLS = 4;
char keys[ROWS][COLS] = {
  { '1', '2', '3', 'A' },
  { '4', '5', '6', 'B' },
  { '7', '8', '9', 'C' },
  { '*', '0', '#', 'D' }
};
byte rowPins[ROWS] = {0, 1, 2, 3};
byte colPins[COLS] = {4, 5, 6, 7};
int I2CADDR = 0x20;
LiquidCrystal_I2C lcd(0x27, 20, 4);
TwoWire *jwire = &Wire;
Keypad_I2C kpd(makeKeymap(keys), rowPins, colPins,
ROWS, COLS, I2CADDR, PCF8574, jwire);
#define RST_PIN 15
#define SS_PIN 5
MFRC522 mfrc522(SS_PIN, RST_PIN);
#define RFID mfrc522.PICC_IsNewCardPresent() &&
mfrc522.PICC_ReadCardSerial()
CountUpDownTimer TIMER(DOWN, HIGH);
const char *ssid = "iPhone";
const char *password = "12345678890"
#ifdef Hosting
const char* server = "http://xxxxxxxxxxxxx.com";
const char* path = "/komunikasi-device.php";
```

```

#else
const char* server = "http://172.20.10.2";
const char* path = "/studio/komunikasi-device.php";
#endif
String inStr, dataIn, Data[10];
boolean parsing = false;
int Li = 20;
int Lii = 0;
unsigned long prevmillis = 0;
char str_no[10] = {0};
int indx_no = 0;

#define pinBuzzer 12
#define RELAY_PIN 13
#define ON 0
#define OFF 1

#define EEPROM_SIZE 64
struct {
    bool INIT_DEVICE;
    String UID;
void setup() {
    Serial.begin(9600);
    Serial2.begin(9600);
    SPI.begin();
    // lcd.init();
    lcd.begin();
    pinMode(pinBuzzer, OUTPUT);
    pinMode(RELAY_PIN, OUTPUT);
    digitalWrite(pinBuzzer, LOW);
    drive_relay(RELAY_PIN, OFF);
    jwire->begin( );
    kpd.begin();

    lcd.setBacklight(3);
    lcd.setCursor(0, 0);
    lcd.print("Initialize.....");
    delay(1000);
    lcd.clear();
    Serial.print("Connecting to ");
    Serial.println(ssid);
    WiFi.mode(WIFI_STA);
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {

```

```

lcd.setCursor(0, 0); lcd.print("Connecting");
lcd.setCursor(0, 1); lcd.print("WiFi");
for (int i = 4; i <= 20; i++) {
  Serial.print(".");
  delay(150);
  lcd.setCursor(i, 1); lcd.print('.');
  if (WiFi.status() != WL_CONNECTED) {
    if (i == 19) {
      lcd.setCursor(5, 1); lcd.print("");
      i = 4;
    }
  }
}
}
printWifiStatus();
Serial.println("");
lcd.clear();
mfr522.PCD_Init();
mfr522.PCD_DumpVersionToSerial();
mp3_set_serial (Serial2);
mp3_set_volume (30);
if (!EEPROM.begin(EEPROM_SIZE)) {
  Serial.println("failed to initialise EEPROM");
  ESP.restart();
}
EEPROM.commit();
EEPROM.get(0, MEMO); // get data eeprom
if (!MEMO.INIT_DEVICE) {
  MEMO.INIT_DEVICE = true;
  MEMO.UID = "";
  EEPROM.put(0, MEMO);
  EEPROM.commit();
  delay(200);
}

Serial.println("");
dataIn = "";
}
void loop() {
  if (WiFi.status() == WL_CONNECTED) {
    lcd.setCursor(2, 0); lcd.print("' STUDIO RENTAL
  ');
    lcd.setCursor(0, 2); lcd.print(" PESAN[A]
  [B]SALDO");
    char key = kpd.getKey();

```

```

switch (key) {
    case 'A' : order(); break;
    case 'B' : balance(); break;
    case 'C' : topup(); break;
    case 'D' : registration(); break;
}
}
else if (WiFi.status() != WL_CONNECTED) {
    Serial.println("WiFi Disconnected, RESET !");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("WiFi Disconnect");
    lcd.setCursor(0, 1);
    lcd.print("Reset.. ");
    delay(2000);
    ESP.restart();
}

//-----
PROGRAM VOID ORDER
-----//

int duration;
void order() {
    lcd.clear();
    delay(200);
    Clear_Scroll_LCD_Left();
    lcd.setCursor(0, 0); lcd.print("Loading...");
    delay(350);
    lcd.clear();
    bool ok = false;
    bool exitt = false;
    String UID = "";
    double pay_amount = 0;
    int min_duration = 1;
    duration = min_duration;
    double price_hour;
    int httpCode = httpRequest("", "set", "");
    if (httpCode == 200) price_hour = Data[2].toFloat();
    if (httpCode != 200 || price_hour == 0) exitt =
true;
    while (!exitt) {
        pay_amount = duration * price_hour;
        String units_t;
        if (TIMER_SYS == HOUR) units_t = "Jam";

```

```

if (TIMER_SYS == MINUTE) units_t = "Menit";
    if (TIMER_SYS == SECOND) units_t = "Detik";
        lcd.setCursor(0, 1); lcd.print("> ");
lcd.print(duration); lcd.print((String) " " +
units_t);
    lcd.print(" [Rp "); lcd.print(rupiah(payment));
lcd.print(" ");
    lcd.setCursor(19, 1); lcd.print("]");
    lcd.setCursor(0, 2); lcd.print("> Rp ");
lcd.print(rupiah(price_hour)); lcd.print((String) "/"
+ units_t);
    lcd.setCursor(1, 3); lcd.print("[*] BACK");
    lcd.setCursor(10, 3); lcd.print("#] TOPUP");
    String text = "Masukkan durasi sewa lalu scan
kartu";
    lcd.setCursor(0, 0);
    if (millis() - prevmillis >= 250) {
        lcd.print(Scroll_LCD_Left(text));
        prevmillis = millis();
    }
    if (RFID) {
        load();
        UID = uid();
        int httpCode = httpRequest(UID, "pay",
(String)duration);
        lcd.clear();
        String nama = Data[3];
        String bayar = Data[4];
        String sisa_saldo = Data[5];
        String status_saldo = Data[6];
        if (httpCode == 200) {
            Clear_Scroll_LCD_Left();
            if (nama.length() <= 20) {
                lcd.setCursor(0, 0); lcd.print(nama);
            } else text_run(nama, 200, 0, 0);
            if (Data[2] == "idsudah" && status_saldo ==
"Saldo Mencukupi") {
                String text = "Bayar Rp " + bayar;
                String text1 = "Saldo Rp " + sisa_saldo;
                lcd.setCursor(0, 2); lcd.print(text);
                lcd.setCursor(0, 3); lcd.print(text1);
                delay(3000);
                MEMO.UID = UID;
                EEPROM.put(0, MEMO);
                EEPROM.commit();
            }
        }
    }
}

```

```

if (TIMER_SYS == HOUR) TIMER.SetTimer(duration, 0, 0);
    if (TIMER_SYS == MINUTE) TIMER.SetTimer(0,
duration, 0);
    if (TIMER_SYS == SECOND) TIMER.SetTimer(0,
0, duration);
    // TIMER.SetTimer(0, 12, 0);
    TIMER.StartTimer();
    timer_display();
    exitt = true;
} else if (Data[2] == "idsudah" &&
status_saldo == "Tidak Cukup") {
    String text = "Saldo Rp " + sisa_saldo;
    lcd.setCursor(0, 2); lcd.print("Saldo Tidak
Cukup");
    lcd.setCursor(0, 3); lcd.print(text);
    buzz(100, 80);
    buzz(80, 100);
    buzz(100, 80);
    delay(2000);
} else {
    lcd.clear();
    lcd.setCursor(0, 0); lcd.print("Belum
Terdaftar!");
    buzz(100, 80);
    buzz(80, 100);
    buzz(100, 80);
}
} else {
    lcd.clear();
    lcd.setCursor(0, 0); lcd.print("Coba Lagi!!");
    buzz(100, 80);
    buzz(80, 100);
    buzz(100, 80);
    delay(1000);
    lcd.clear();
}
delay(1000);
lcd.clear();
}

char key = kpd.getKey();
if (key) {
    switch (key) {
        case NO_KEY: break;

```



```

buzz(100, 0);
    mp3_play (1);
} else if (TIMER.ShowHours() == 0 &&
TIMER.ShowMinutes() == 0 && TIMER.ShowSeconds() == 30)
{
    buzz(100, 0);
    mp3_play (2);
} else if (TIMER.ShowHours() == 0 &&
TIMER.ShowMinutes() == 0 && TIMER.ShowSeconds() == 0)
{
    buzz(100, 80);
    buzz(80, 100);
    buzz(100, 80);
    mp3_play (3);
    delay(10000);
    break;
}
}
char key = kpd.getKey();
if (key == '*' && !selesai) {
    selesai = true;
    lcd.setCursor(0, 3); lcd.print("Scan Card!
[#]CANCEL");
} else if (key == '#' && selesai) {
    selesai = false;
    lcd.setCursor(0, 3); lcd.print("[*]
SELESAI      ");
}

if (RFID && selesai) {
    // Serial.println((String)"UID tag :" + uid());
    if (uid() == MEMO.UID) break;
}
delay(1);
}
drive_relay(RELAY_PIN, OFF);
lcd.clear();
lcd.setCursor(0, 0); lcd.print("BOOKING SELESAI..");
lcd.setCursor(3, 2); lcd.print("TERIMA KASIH..");
delay(2000);
}

```

```

//-----
PROGRAM VOID CEK SALDO
-----//

void balance() {
    lcd.clear();
    delay(200);
    lcd.setCursor(0, 0); lcd.print("Loading...");
    delay(350);
    lcd.clear();
    String UID = "";
    String nama;
    double saldo;
    while (1) {
        lcd.setCursor(2, 0); lcd.print("- CHECK SALDO -");
        lcd.setCursor(2, 1); lcd.print("Put Card to Scan");
        lcd.setCursor(0, 3); lcd.print("[*] CANCEL");

        if (RFID) {
            load();
            UID = uid();
            int httpCode = httpRequest(UID, "bal", "");
            lcd.clear();
            if (httpCode == 200) {
                nama = Data[3];
                saldo = Data[4].toFloat();

                if (Data[2] == "idsudah") {
                    if (nama.length() <= 20) {
                        lcd.setCursor(0, 0); lcd.print(nama);
                    } else text_run(nama, 200, 0, 0);
                    lcd.setCursor(0, 1); lcd.print(" > Rp ");
                    lcd.print(rupiah(saldo));
                    delay(4000);
                    break;
                } else {
                    lcd.setCursor(0, 0); lcd.print("Belum Terdaftar!");
                    buzz(100, 80);
                    buzz(80, 100);
                    buzz(100, 80);
                    delay(2000);
                    lcd.clear();
                }
            }
        }
    }
}

```

```

    }
    } else {
        lcd.clear();
        lcd.setCursor(0, 0); lcd.print("Coba Lagi!!");
        buzz(100, 80);
        buzz(80, 100);
        buzz(100, 80);
        delay(1000);
        return;
    }
}

char key = kpd.getKey();
if (key == '*') break;
}
lcd.clear();
delay(200);
}

//-----
PROGRAM VOID TOP UP
-----//

void topup() {
    lcd.clear();
    delay(200);
    lcd.setCursor(0, 0); lcd.print("Loading...");
    delay(350);
    lcd.clear();
    String UID = "";
    String nama;
    double saldo;
    while (1) {
        lcd.setCursor(2, 0); lcd.print("- TOP UP SALDO -");
    };
    lcd.setCursor(2, 1); lcd.print("Put Card to");
    lcd.setCursor(0, 3); lcd.print("[*] CANCEL");

    if (RFID) {
        load();
        UID = uid();
        int httpCode = httpRequest(UID, "topup", "");
        lcd.clear();
    }
}

```

```

if (httpCode == 200) {
    nama = Data[3];
    saldo = Data[4].toFloat();
    if (Data[2] == "idsudah") {
        if (nama.length() <= 16) {
            lcd.setCursor(0, 0); lcd.print(nama);
        } else text_run(nama, 200, 0, 0);
        lcd.setCursor(0, 0); lcd.print(nama);
        lcd.setCursor(0, 1); lcd.print(" > Rp ");
    lcd.print(rupiah(saldo));
        delay(4000);
        break;
    } else {
        lcd.setCursor(0, 0); lcd.print("Belum
Terdaftar!");
        buzz(100, 80);
        buzz(80, 100);
        buzz(100, 80);
        delay(2000);
        lcd.clear();
    }
} else {
    lcd.clear();
    lcd.setCursor(0, 0); lcd.print("Coba Lagi!!");
    buzz(100, 80);
    buzz(80, 100);
    buzz(100, 80);
    delay(1000);
    return;
}
}

char key = kpd.getKey();
if (key == '*') break;
}
lcd.clear();
delay(200);
}
//-----
PROGRAM REGISTRASI KARTU
-----//

void registration() {
    lcd.clear();
    delay(200);
}

```

```

lcd.setCursor(0, 0); lcd.print("Loading...");
delay(350);
lcd.clear();
String nama;
while (1) {
    lcd.setCursor(3, 0); lcd.print("- MEMBER UID -");
    lcd.setCursor(2, 1); lcd.print("Put Card to
Scan");
    lcd.setCursor(0, 3); lcd.print("[*] DONE");

    if (RFID) {
        load();
        int httpCode = httpRequest(uid(), "regis", "");
        lcd.clear();

        if (httpCode == 200) {
            nama = Data[3];
            if (Data[2] == "idsudah") {
                lcd.setCursor(0, 0); lcd.print("Sudah
Terdaftar!")
            } else {
                lcd.clear();
                lcd.setCursor(0, 0); lcd.print("Belum
Terdaftar!");
                lcd.setCursor(0, 1); lcd.print("Silahkan
melakukan");
                lcd.setCursor(0, 2); lcd.print("input
data..");
            }
            delay(2000);
        } else {
            lcd.clear();
            lcd.setCursor(0, 0); lcd.print("Coba Lagi!!");
            buzz(100, 80);
            buzz(80, 100);
            buzz(100, 80);
            delay(1000);
        }
        lcd.clear();
    }
    char key = kpd.getKey();
    if (key == '*') break;
}
lcd.clear();
delay(200);

```

```

String uid() {
    String CARD_ID;
    CARD_ID = "";
    for (byte i = 0; i < mfr522.uid.size; i++) {
        CARD_ID.concat(String(mfr522.uid.uidByte[i] <
0x10 ? " 0" : " "));
        CARD_ID.concat(String(mfr522.uid.uidByte[i],
HEX));
    }
    CARD_ID.toUpperCase();
    Serial.print("UID tag :"); Serial.println(CARD_ID);
    mfr522.PICC_HaltA();
    mfr522.PCD_StopCryptol();
    buzz(100, 0);
    buzz(0, 100);
    return CARD_ID;
}
void load() {
    lcd.clear();
    lcd.setCursor(0, 0); lcd.print("Processing..");
    delay(200);
}
int httpRequest(String _uid, String _mode, String
_val) {
    HTTPClient http;

    String postData;
    if (_mode == "topup") {
        postData += "cardid=" + _uid;
        postData += "&mode=" + _mode;
        postData += "&val=" + _val;
    } else if (_mode == "pay") {
        postData += "cardid=" + _uid;
        postData += "&mode=" + _mode;
        postData += "&val=" + _val;
    } else {
        postData += "cardid=" + _uid;
        postData += "&mode=" + _mode;
    }
    http.begin((String)server + path);
    http.addHeader("Content-Type", "application/x-www-
form-urlencoded");
    int httpCode = http.POST(postData);
    String payload = http.getString();
}

```

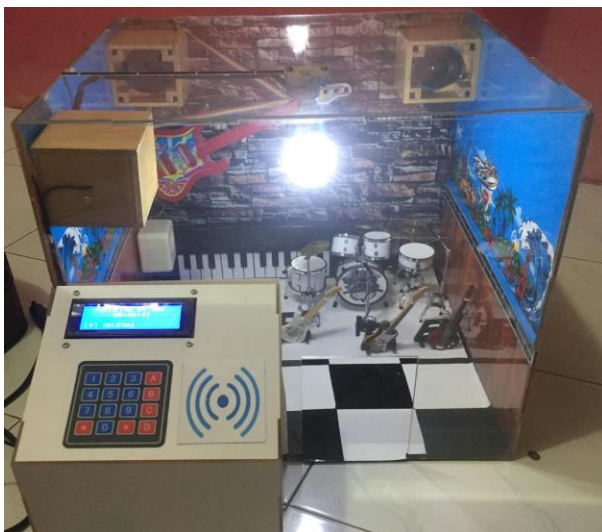
```

Serial.println(httpCode);
Serial.println(payload);
String Splitpayload = payload.substring(2);
// Terima data untuk di parsing -----
if (payload != "") {
    for (int i = 0; i <= Splitpayload.length(); i++) {
        inStr = Splitpayload.substring(i, (i + 1));
        dataIn += inStr;
        if (inStr == "#") parsing = true;
    }
    if (parsing) {
        parsingData();
        parsing = false;
        dataIn = "";
    }
}
http.end();

return httpCode;
}

```


LAMPIRAN B



BIODATA PENULIS



Nama : Erik Miftachul
Tempat/Tanggal Lahir : Kendari, 04 Februari 1998
Alamat : Jl. Budi Utomo Blok E No.2,
Kadia,Kendari,Sulawesi Tenggara
Telepon/Hp : 082244416861
Hobi : Bulu Tangkis
Email : Erikm435@gmail.com
Riwayat Pendidikan :
• SDN 16 Baruga : 2004 - 2010
• SMPN 17 Kendari : 2010 - 2013
• SMKN 2 Kendari : 2013 - 2016
• Politeknik Negeri Cilacap : 2019 - 2022