

LAMPIRAN

Lampiran 1. Program Kontroller Keseluruhan

Count_menu.ino

```
#include <EEPROM.h>
#include <SPI.h>
#include <MFRC522.h>
#include <LiquidCrystal.h>
#include <DS3231.h>
#include <EEPROM.h>
#include <TimerOne.h>

#define start 3
//#define stop 2
#define up A0
#define ok A1
#define down A2
#define pin_tim1 0
#define pin_interlock 1
#define add_count 1022
#define add_timer1 1018
#define add_max_count 987
#define SS_PIN 10
#define RST_PIN 2
#define data1 "234b71c"
#define data2 "234b71d"
#define data3 "234b71e"

LiquidCrystal lcd(9, A3, 7, 6, 5, 4);
MFRC522 mfrc522(SS_PIN, RST_PIN); // Instance of the class

DS3231 Clock;
RTCLib rtc;
DateTime now;

int state_menu;
const char hari[][7] = {"Senin", "Selasa", "Rabu", "Kamis", "Jum'at", "Sabtu",
"Minggu", "Senin"};
uint16_t yy;
byte dd, mm, h, m, s, dow;
char buff_time_akhir[15];
unsigned long int counting;
unsigned int jam, menit, detik;
```

```

uint8_t pilihan;
bool start_count;
byte menu_ok;

uint16_t read_add;
uint16_t add_ee_write;
uint16_t add_ee_read;
uint8_t idx_cnt;

boolean flag_lanjut;
String uid;

void setup() {
  lcd.begin(16, 2);
  Wire.begin();
  pinMode(start,INPUT_PULLUP);
  //pinMode(stop,INPUT_PULLUP);
  pinMode(up,INPUT_PULLUP);
  pinMode(ok,INPUT_PULLUP);
  pinMode(down,INPUT_PULLUP);
  pinMode(pin_tim1,OUTPUT);
  //pinMode(pin_tim2,OUTPUT);
  pinMode(pin_interlock,OUTPUT);
  SPI.begin(); // Init SPI bus
  mfrc522.PCD_Init(); // Init MFRC522

  //Serial.begin(9600);
}

void loop()
{
  //counter();
  menu();
  utama();
}

```

Function.ino

```

void kembali()
{
  menu_ok = 0;
  //pilihan=0;
  utama();
}

void tekan_ok()
{

```

```

if (digitalRead(ok) == LOW)
{
  while (digitalRead(ok) == LOW) {}
  menu_ok++;
  lcd.clear();
}
//Serial.println(menu_ok);
}

void tampil_jam()
{
  char buff_time_now[17];
  now = rtc.now();
  h = now.hour();
  m = now.minute();
  s = now.second();
  sprintf(buff_time_now, "Jam: %02d:%02d:%02d ", h, m, s);
  lcd.setCursor(0, 0);
  lcd.print(buff_time_now);
}

void tampil_tanggal()
{
  char buff_date_now[17];
  now = rtc.now();
  yy = Clock.getYear();
  mm = now.month();
  dd = now.day();
  dow = Clock.getDoW();
  sprintf(buff_date_now, "%s, %02d-%02d-%02d ", hari[dow], dd, mm, yy);
  lcd.setCursor(0, 1);
  lcd.print(buff_date_now);
}

void counter()
{
  if(start_count==true)
  {
    char buff_time_now[15];
    char buff_count[15];
    long unsigned int mod;
    static uint8_t cs;
    now = rtc.now();
    yy = now.year();
    mm = now.month();
    dd = now.day();
  }
}

```

```

h = now.hour();
m = now.minute();
s = now.second();

sprintf(buff_time_now,"Jam: %02d:%02d:%02d  ", h, m, s);
lcd.setCursor(0, 0);
lcd.print( buff_time_now);

jam = counting/3600;
mod = counting%3600;
menit = mod/60;
detik = mod%60;
if(cs!=s)
{
    counting++;
    cs=s;
}
idx_cnt = add_ee_write/21;
sprintf(buff_count,"Cnt%02d: %02d:%02d:%02d ", idx_cnt, jam, menit, detik);
lcd.setCursor(0, 1);
lcd.print( buff_count );
//Serial.print(start_count);

}

if(!start_count)
{
    if(add_ee_write==add_max_count)
    {
        tampil_jam();
        lcd.setCursor(0, 1);
        lcd.print("Memori Penuh!!..");
    }
    else
    {
        tampil_jam();
        tampil_tanggal();
    }
}
}

void counter_stop()
{
    char buff_date_time[16];
    now = rtc.now();
    yy = Clock.getYear();

```

```

mm = now.month();
dd = now.day();
h = now.hour();
m = now.minute();
sprintf(buff_date_time,"%02d:%02d, %02d/%02d/%02d", h, m, dd, mm, yy);
EEPROM.put(add_ee_write, counting);
EEPROM.put((add_ee_write+4), buff_date_time);
if(add_ee_write<add_max_count)
{
    add_ee_write+=21;
}
EEPROM.put(add_count, add_ee_write);
counting=0;
}

void read_eeprom(uint16_t add_ee_read)
{
    unsigned long int read_count;
    char buff_count[15];
    char buff_date_time[16];
    EEPROM.get(add_ee_read, read_count);
    EEPROM.get((add_ee_read+4), buff_date_time);
    jam = read_count/3600;
    menit = (read_count%3600)/60;
    detik = (read_count%3600)%60;
    lcd.clear();
    lcd.setCursor(0, 0);
    if(buff_date_time[0]!='\0')
    {
        lcd.print("Counter Kosong");
    }
    else
    {
        lcd.print(buff_date_time);
    }
    idx_cnt=add_ee_read/21;
    sprintf(buff_count,"Cnt%02d: %02d:%02d:%02d", idx_cnt , jam, menit, detik);
    lcd.setCursor(0, 1);
    lcd.print( buff_count );
}

```

Main.ino

```

void utama()
{
    digitalWrite(pin_tim1, LOW);
    //digitalWrite(pin_tim2, LOW);
}

```



```

digitalWrite(pin_interlock, LOW);
while(menu_ok==0)
{
  if ( mfr522.PICC_IsNewCardPresent())
  {
    if ( mfr522.PICC_ReadCardSerial())
    {
      //Serial.print("Tag UID:");
      for (byte i = 0; i < mfr522.uid.size; i++)
      {
        uid = uid + String (mfr522.uid.uidByte[i], HEX);
      }
      lcd.clear();
      while(1)
      {
        if((uid!=data1))
        {
          lcd.setCursor(0, 0);
          lcd.print("Maaf Anda");
          lcd.setCursor(0, 1);
          lcd.print("Tidak Terdaftar");
          delay(3000);
          break;
        }
        else
        {
          lcd.setCursor(0, 0);
          lcd.print("Anda Terdaftar");
          lcd.setCursor(0, 1);
          lcd.print("Tekan Start");
        }
      }
    }
  }
  if((digitalRead(start)==LOW) and (start_count==false))
  {
    if(flag_lanjut==false)
    {
      EEPROM.get(add_count, read_add);
      add_ee_write=read_add;
    }
    else
    {
      flag_lanjut=false;
    }
  }
}

```

```

        //while(digitalRead(start)==LOW){}
        start_count=true;
        digitalWrite(pin_interlock, HIGH);
        break;
    }
}
uid="";
mfr522.PICC_HaltA();
}
}

if((digitalRead(start)==HIGH)&&(start_count==true))
{
    digitalWrite(pin_interlock, LOW);
    counter_stop();
    lcd.clear();
    start_count=false;
    //while(digitalRead(stop)==LOW){}
}
counter();
tekan_ok();
}
}
}

```

Menu.ino

```

void menu()
{
    while (menu_ok == 1)
    {
        tekan_ok();
        if (digitalRead(up) == LOW)
        {
            if(pilihan>=4)
            {
                pilihan=0;
            }
            else
            {
                while (digitalRead(up) == LOW) {}
                pilihan++;
                lcd.clear();
            }
            delay(10);
        }
    }
}

```

```

if (digitalRead(down) == LOW)
{
  if (pilihan<= 0)
  {
    pilihan=4;
  }
  else
  {
    while (digitalRead(down) == LOW) {}
    pilihan--;
    lcd.clear();
  }
  delay(10);
}
lcd.setCursor(0, 0);
lcd.print("Pilih Menu: ");
tekan_ok();

```

Read_count.ino

```

void pilih_read_eeprom()
{
  if (menu_ok == 2)
  {
    if(add_ee_write>0)
    {
      add_ee_read = add_ee_write-=21;
    }
    read_eeprom(add_ee_read);
  }
  while (menu_ok == 2)
  {
    tekan_ok();
    if (digitalRead(up) == LOW)
    {
      while (digitalRead(up) == LOW) {}
      if (add_ee_read < add_max_count)
      {
        add_ee_read += 21;
        read_eeprom(add_ee_read);
      }
    }
  }

  else if (digitalRead(down) == LOW)
  {
    while (digitalRead(down) == LOW) {}

```



```

    if (add_ee_read > 0)
    {
        add_ee_read -= 21;
        read_eeprom(add_ee_read);
    }
}

else if (digitalRead(start) == LOW)
{
    while (digitalRead(start) == LOW) {}
    lanjut();
}

else if (menu_ok > 2)
{
    kembali();
}

}
}

```

Reset.ino

```

void reset_ee()
{
    bool rst=false;
    uint8_t cnt=0;
    while (menu_ok == 2)
    {
        tekan_ok();
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("5.Yakin Reset?");
        lcd.setCursor(0, 1);
        lcd.print("Tidak");
        while(cnt<1)
        {
            if (digitalRead(ok) == LOW)
            {
                while (digitalRead(ok) == LOW) {}
                cnt++;
                lcd.clear();
            }

            else if (digitalRead(up) == LOW)
            {
                while (digitalRead(up) == LOW) {}
            }
        }
    }
}

```

```

if (rst==true)
{
  rst=false;
  lcd.setCursor(0, 1);
  lcd.print("Tidak");
}
else
{
  rst=true;
  lcd.setCursor(0, 1);
  lcd.print("Ya ");
}
}
}

```

```

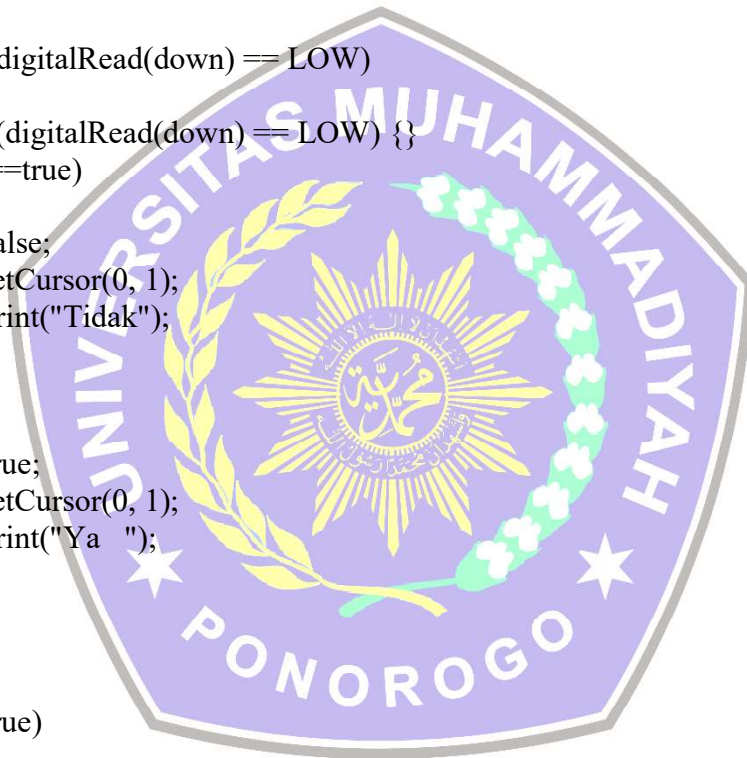
else if (digitalRead(down) == LOW)
{
  while (digitalRead(down) == LOW) {
    if (rst==true)
    {
      rst=false;
      lcd.setCursor(0, 1);
      lcd.print("Tidak");
    }
    else
    {
      rst=true;
      lcd.setCursor(0, 1);
      lcd.print("Ya ");
    }
  }
}
}

```

```

if(rst==true)
{
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Reseting...");
  for (uint16_t i = 0; i < 1024; i++)
  {
    EEPROM.put(i, 0);
    delay(5);
  }
  add_ee_write = 0;
  add_ee_read = 0;
  flag_lanjut = true;
  lcd.clear();
}

```



```
kembali();
}
kembali();
}
}
```

Timer.ino

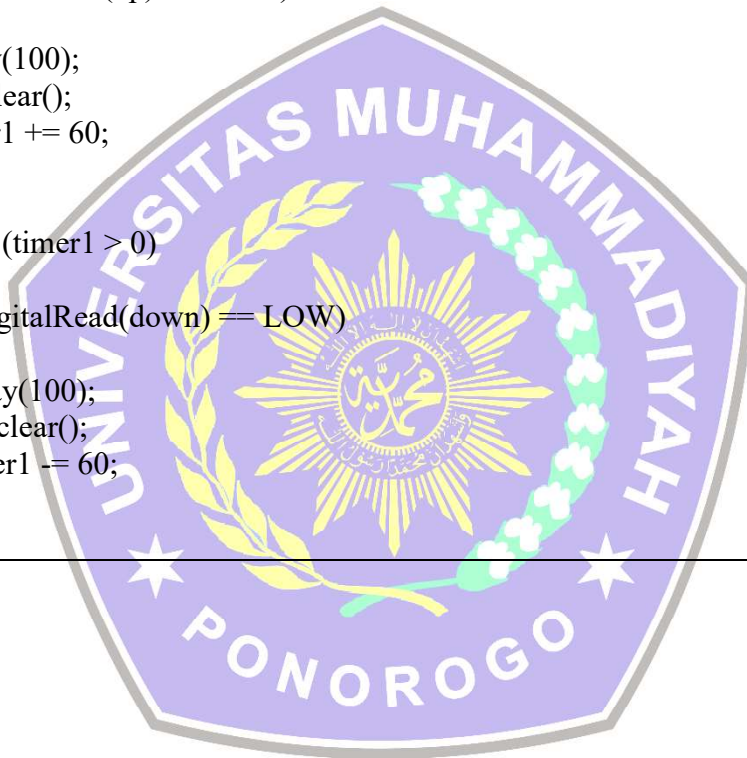
```
void timer_1()
{
  char buff_timer1[16];
  uint8_t jam, menit, detik, cnt = 0;
  unsigned long int durasi_tim1;
  unsigned long int timer1;
  while (menu_ok == 2)
  {
    tekan_ok();
    lcd.setCursor(0, 1);
    sprintf(buff_timer1, "Timer1: %02d:%02d:%02d", jam, menit, detik);
    lcd.print(buff_timer1);
    while (cnt < 3)
    {
      if (digitalRead(ok) == LOW)
      {
        while (digitalRead(ok) == LOW) {}
        cnt++;
        durasi_tim1=timer1;
        lcd.clear();
      }
      if (cnt == 0)
      {
        lcd.setCursor(0, 0);
        lcd.print("2.Timer1 Jam");
        if (digitalRead(up) == LOW)
        {
          delay(100);
          lcd.clear();
          timer1 += 3600;
        }
        else if (timer1 > 0)
        {
          if (digitalRead(down) == LOW)
          {
            delay(100);
            lcd.clear();
            timer1 -= 3600;
          }
        }
      }
    }
  }
}
```

```

}
jam = timer1 / 3600;
menit = (timer1 % 3600) / 60;
detik = (timer1 % 3600) % 60;
lcd.setCursor(0, 1);
sprintf(buff_timer1, "Timer1: %02d:%02d:%02d\0", jam, menit, detik);
lcd.print(buff_timer1);
}
else if (cnt == 1)
{
lcd.setCursor(0, 0);
lcd.print("2.Timer1 menit");
if (digitalRead(up) == LOW)
{
delay(100);
lcd.clear();
timer1 += 60;
}

else if (timer1 > 0)
{
if (digitalRead(down) == LOW)
{
delay(100);
lcd.clear();
timer1 -= 60;
}
}
}
}

```



Lampiran 2 Dokumentasi Pengujian Data

