



**LAMPIRAN**

# LAMPIRAN

## Lampiran A. Code Arduino

### 1. Main Code

```
/*
 * pemanggilan library dan juga inialisasi variable
 * yang berhubungan dengan wifi dan telegram
 */
#ifdef ESP32
#include <WiFi.h>
#else
#include <ESP8266WiFi.h>
#endif
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>
const char* ssid = "ibe-store";
const char* password = "nahkanketauanmake412";
#define BOTtoken
"1569696618:AAEkvMcEmXWnooa2AoOVHJ43u
Z3Sk5sf8w" //Token/API botFather
#define CHAT_ID "971336358" //@idbot
#ifdef ESP8266
X509List
cert(TELEGRAM_CERTIFICATE_ROOT);
#endif
WiFiClientSecure client;
UniversalTelegramBot bot(BOTtoken, client);
/*=====*/
/*
 * pemanggilan library dan juga inialisasi variable
 * yang berhubungan dengan lcd, tombol, buzzer, dan
 * rotary
 */
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
#define OK 14
#define UP 12
#define DOWN 13
#define rotary_pin 33
#define buzzer 23
long pulses = 0;
float jarak = 0;
float keliling = 34.56;
float lubang = 20;
void IRAM_ATTR counter_rotary()
{
pulses += 1;
}
String satuan_waktu = " Bulan";
String satuan = " Km";
void bacaJarak(){ //fungsi untuk membaca jarak
if(satuan == " Km"){
jarak = (pulses/15) * keliling / 100000;
}
else if(satuan == " M"){
jarak = (pulses/15) * keliling / 100;
}
//jarak = ((pulses/15) * keliling) / 100;
Serial.print("Pulse:
");Serial.print(pulses,0);Serial.print(" ");
Serial.print("Jarak: ");Serial.println(jarak);
}

/*****
 * NTP
 *****/
#include <NTPClient.h>
#include <WiFiUdp.h>
const long utcOffsetInSeconds = 25200;
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "asia.pool.ntp.org",
utcOffsetInSeconds);
/*=====*/
void setup() {
Serial.begin(9600);
pinMode(OK, INPUT_PULLUP);
pinMode(UP, INPUT_PULLUP);
pinMode(DOWN, INPUT_PULLUP);
pinMode(buzzer, OUTPUT);
buzzer_(1, 75);
lcd.begin();
lcd.setCursor(0,0);
loading();
lcd.clear();
pinMode(rotary_pin, INPUT);
attachInterrupt(rotary_pin, counter_rotary,
RISING);

#ifdef ESP8266
configTime(0, 0, "pool.ntp.org");
client.setTrustAnchors(&cert);
#endif
WiFi.mode(WIFI_STA);
WiFi.begin(ssid, password);
#ifdef ESP32

client.setCACert(TELEGRAM_CERTIFICATE_R
OOT);
#endif
while (WiFi.status() != WL_CONNECTED) {
delay(1000);
lcd.setCursor(0,0);
lcd.print("Connecting to...");
lcd.setCursor(0,1);
lcd.print(ssid);
Serial.println("Connecting to WiFi.");
}
lcd.clear();
Serial.println(WiFi.localIP());
lcd.setCursor(0,0);
lcd.print("Connected...");
lcd.setCursor(0,1);
lcd.print("IP:");
lcd.print(WiFi.localIP());
delay(2500);
lcd.clear();
buzzer_(3, 75);
timeClient.begin();
}
int limit_jarak = 0, temp_limit_jarak = 0;
int limit_waktu = 0, temp_limit_waktu = 0;
int delay_btn = 500;
long temp_millis = 0;
```

```

int idx_posisi = 0; //0.menu 1.setting 2.mulai
int point_menu = 1; //1.setting 2.mulai
bool notif_jarak_half = false;
bool notif_waktu_half = false;
int waktu = 0;
int temp_waktu = 0;
int cunter = 0;
void loop() {
    if(idx_posisi == 0){ //kondisi saat pertama kali
dihidupkan
        menu_depan(); //memanggil fungsi menu_depan
yang berisi menu pemilihan setting dan mulai
    }
    else if(idx_posisi == 1){
        menu_setting(); //memanggil menu setting yang
berisi set limit dan set satuan
    }
    else if(idx_posisi == 2){ //kondisi saat pemilihan
mode mulai
        if(millis() - temp_millis > 1000){ //setiap satu
ddetik memanggil fungsi baca jarak dan
menampilkan jarak di baris ke 2 lcd
            temp_millis = millis();
            cunter += 1;
            if(cunter > 10000){
                cunter = 0;
            }
            bacaJarak();
            if(cunter % 2 == 0){
                lcd.setCursor(0,0);
                lcd.print("Lim: "); //menampilkan limit waktu
di baris pertama
                lcd.print(limit_waktu);
                lcd.print(satuan_waktu);
                lcd.setCursor(0,1);
                lcd.print("Now: ");
                lcd.setCursor(5,1);
                lcd.print(" ");
                lcd.setCursor(5,1);
                lcd.print(waktu-1);
                lcd.print(satuan_waktu);
            }else{
                lcd.setCursor(0,0);
                lcd.print("Lim: "); //menampilkan limit jarak di
baris pertama
                lcd.print(limit_jarak);
                lcd.print(satuan);
                lcd.setCursor(0,1);
                lcd.print("Now: ");
                lcd.setCursor(5,1);
                lcd.print(" ");
                lcd.setCursor(5,1);
                lcd.print(jarak);
                lcd.print(satuan);
            }
            if(jarak > limit_jarak){ //jika jarak > limit kirim
notif ke telegram
                bot.sendMessage(CHAT_ID, "Jarak yang
ditempuh sudah melebihi limit " +
(String)limit_jarak + satuan, "");
                lcd.clear();
                lcd.setCursor(0,0);
                lcd.print("WARNING !!!");
                for(int i=0; i<10; i++){
                    lcd.setCursor(0,1);
                    lcd.print(". Jarak > Limit!");
                    digitalWrite(buzzer, HIGH);
                    delay(500);
                    lcd.setCursor(0,1);
                    lcd.print(" ");
                }
                digitalWrite(buzzer, LOW);
                delay(500);
            }
        }
    }
    else if(jarak > limit_jarak/2 and
!notif_jarak_half){
        bot.sendMessage(CHAT_ID, "Jarak yang
ditempuh sudah melebihi 1/2 limit " +
(String)limit_jarak + satuan, "");
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("WARNING !!!");
        for(int i=0; i<3; i++){
            lcd.setCursor(0,1);
            lcd.print("Jarak > 1/2 Lmit");
            digitalWrite(buzzer, HIGH);
            delay(500);
            lcd.setCursor(0,1);
            lcd.print(" ");
            digitalWrite(buzzer, LOW);
            delay(500);
        }
        notif_jarak_half = true;
    }

    timeClient.update();
    if(satuan_waktu == " Bulan"){
        time_t epochTime =
timeClient.getEpochTime();
        struct tm *ptm = gmtime ((time_t
*)&epochTime);
        int currentMonth = ptm->tm_mon+1;
        // Serial.print("Month: ");
        // Serial.println(currentMonth);
        if(temp_waktu != currentMonth){
            waktu += 1;
            temp_waktu = currentMonth;
        }
    }
    else if(satuan_waktu == " Detik"){
        if(temp_waktu != timeClient.getSeconds()){
            waktu += 1;
            temp_waktu = timeClient.getSeconds();
        }
    }
    if(waktu > limit_waktu){ //jika jarak > limit
kirim notif ke telegram
        bot.sendMessage(CHAT_ID, "Waktu
penggunaan sudah melebihi Limit " +
(String)limit_waktu + satuan_waktu, "");
    }
}

```

```

lcd.clear();
lcd.setCursor(0,0);
lcd.print("WARNING !!!");
for(int i=0; i<10; i++){
  lcd.setCursor(0,1);
  lcd.print(".Waktu > Limit!");
  digitalWrite(buzzer, HIGH);
  delay(500);
  lcd.setCursor(0,1);
  lcd.print(" ");
  digitalWrite(buzzer, LOW);
  delay(500);
}
lcd.clear();
lcd.setCursor(0,0);
lcd.print("#SILKAHKAN RESET");
while(1){
  if(digitalRead(OK) == LOW){ //menunggu
penekanan reset untuk kembali memulai
perhitungan
    buzzer_(1, 75);
    delay(delay_btn);
    jarak = 0;
    waktu = 0;
    pulses = 0;
    notif_jarak_half = false;
    notif_waktu_half = false;
    lcd.clear();
    break;
  }
}
else if(waktu > limit_waktu/2 and
!notif_waktu_half){
  bot.sendMessage(CHAT_ID, "Waktu
penggunaan sudah melebihi 1/2 Limit " +
(String)limit_waktu + satuan_waktu, "");
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("WARNING !!!");
  for(int i=0; i<3; i++){
    lcd.setCursor(0,1);
    lcd.print("Waktu > 1/2 Lmit");
    digitalWrite(buzzer, HIGH);
    delay(500);
    lcd.setCursor(0,1);
    lcd.print(" ");
    digitalWrite(buzzer, LOW);
    delay(500);
  }
  notif_waktu_half = true;
}
}
if(digitalRead(DOWN) == LOW){
  buzzer_(1, 75);
  delay(delay_btn);
  jarak = 0;
  pulses = 0;
  lcd.clear();
  idx_posisi = 0;
  notif_jarak_half = false;
  notif_waktu_half = false;
}

// int numNewMessages =
bot.getUpdates(bot.last_message_received + 1);
// while(numNewMessages) {
//   Serial.println("got response");
//   handleNewMessages(numNewMessages);

```

```

// numNewMessages =
bot.getUpdates(bot.last_message_received + 1);
// }
}

```

```

void buzzer_(int loop_, int delay_){
  for(int i=0; i<loop_; i++){
    digitalWrite(buzzer, HIGH);
    delay(delay_);
    digitalWrite(buzzer, LOW);
    delay(delay_);
  }
}

```

## 2. LCD Code

```

// setting animation speed (delay between frames)
int x = 150;
// setting time for Man waiting
int y = 5000;
// START setting all the pixel-art
//setting pixels for standing man:
//step 1 of 1:
  byte stepA1[8] =
  {B00011,B00111,B00111,B00111,B00111,B00001,
B00011,B00111};
  byte stepA2[8] =
  {B00001,B00001,B00011,B00011,B00011,B00001,
B00000,B00001};
  byte stepA3[8] =
  {B11111,B11111,B11111,B11111,B11011,B10001,
B10001};
  byte stepA4[8] =
  {B11111,B11111,B11101,B11101,B11111,B11111,
B11111,B11111};
  byte stepA5[8] =
  {B11000,B11100,B11100,B11100,B11000,B10000,
B11000,B11100};
  byte stepA6[8] =
  {B00000,B10000,B10000,B00000,B10000,B11100,
B11000,B10000};
//setting pixels for running man right:
//step 1 of 3:
  byte stepB1[8] =
  {B00011,B00011,B00011,B00011,B00001,B00000,
B00000,B00000};
  byte stepB2[8] =
  {B00000,B00001,B00001,B00011,B00011,B00011,
B00000,B00001};
  byte stepB3[8] =
  {B11111,B11111,B11111,B11111,B11111,B11111,
B11111,B11110};
  byte stepB4[8] =
  {B11111,B11111,B11101,B11101,B11111,B11111,
B11111,B11111};
  byte stepB5[8] =
  {B00000,B00000,B10000,B10000,B10000,B00000,
B10000,B00000};
  byte stepB6[8] =
  {B00000,B11000,B00000,B11000,B11100,B11000,
B10000,B00000};
//step 2 of 3
  byte stepC1[8] =
  {B11111,B11111,B11101,B00011,B00111,B01111,
B01111,B00111};

```

```

byte stepC2[8] =
{B00000,B00001,B00001,B00011,B00011,B00011,
B00000,B00111};
byte stepC3[8] =
{B11111,B11111,B11111,B11111,B11111,B10011,
B00000,B00000};
byte stepC4[8] =
{B11111,B11111,B11101,B11101,B11111,B11111,
B11111,B11111};
byte stepC5[8] =
{B11110,B11110,B01100,B11100,B11100,B11100,
B00000,B00000};
byte stepC6[8] =
{B00000,B11000,B00000,B11000,B11100,B11000,
B10000,B00000};
//step 3 of 3
byte stepD1[8] =
{B00001,B00011,B00111,B00111,B00111,B01111,
B01100,B00000};
byte stepD2[8] =
{B00000,B00000,B00001,B00001,B00011,B00011,
B00011,B00000};
byte stepD3[8] =
{B11111,B11111,B11111,B11111,B11111,B11111,
B11110,B11111};
byte stepD4[8] =
{B00000,B11111,B11111,B11101,B11101,B11111,
B11111,B11111};
byte stepD5[8] =
{B10000,B11000,B10000,B00000,B00000,B00000,
B00000,B00000};
byte stepD6[8] =
{B00000,B00000,B11000,B00000,B11000,B11100,
B11000,B10000};
//setting pixels for man slipping right
//step 1 of 3
byte slipA1[8] =
{B00111,B00111,B00011,B00011,B00001,B01111,
B01111,B00111};
byte slipA2[8] =
{B00000,B00011,B00111,B01111,B01111,B00111,
B00011,B00111};
byte slipA3[8] =
{B11111,B11111,B11111,B11111,B11111,B11111,
B11000,B10000};
byte slipA4[8] =
{B11111,B11111,B11101,B11101,B11111,B11111,
B11111,B11111};
byte slipA5[8] =
{B11000,B11000,B10000,B00000,B00000,B00000,
B00000,B00000};
byte slipA6[8] =
{B00000,B11000,B00000,B11100,B11100,B11000,
B10000,B11000};
//step 2 of 3
byte slipB1[8] =
{B00011,B00011,B00001,B00001,B00000,B00111,
B00111,B00011};
byte slipB2[8] =
{B00000,B00001,B00011,B00111,B00111,B00011,
B00001,B00011};
byte slipB3[8] =
{B11111,B11111,B11111,B11111,B11111,B11111,
B11100,B11000};
byte slipB4[8] =
{B11111,B11111,B11110,B11110,B11111,B11111,
B11111,B11111};
byte slipB5[8] =
{B11100,B11100,B11000,B10000,B10000,B00000,
B00000,B00000};
byte slipB6[8] =
{B00000,B11000,B10000,B11100,B11110,B11100,
B11000,B11100};
//step 3 of 3
byte slipC1[8] =
{B00001,B00001,B00000,B00000,B00000,B00011,
B00011,B00001};
byte slipC2[8] =
{B00000,B00000,B00001,B00011,B00011,B00001,
B00000,B00001};
byte slipC3[8] =
{B11111,B11111,B11111,B11111,B11111,B11111,
B11110,B11100};
byte slipC4[8] =
{B01111,B11111,B11111,B11111,B11111,B11111,
B11111,B11111};
byte slipC5[8] =
{B11110,B11110,B11100,B11000,B11000,B10000,
B00000,B00000};
byte slipC6[8] =
{B10000,B11100,B11000,B11110,B11111,B11110,
B11000,B11110};
//setting pixels for man running left
//step 1 of 3
byte stepE1[8] =
{B00000,B00001,B00001,B00000,B00000,
B00001,B00000};
byte stepE2[8] =
{B00000,B00011,B00000,B00111,B00011,
B00001,B00000};
byte stepE3[8] =
{B11111,B11111,B11111,B11111,B11111,B11111,
B11111,B01111};
byte stepE4[8] =
{B11111,B11111,B10111,B10111,B11111,B11111,
B11111,B11111};
byte stepE5[8] =
{B11000,B11000,B11000,B11000,B10000,B00000,
B00000,B00000};
byte stepE6[8] =
{B00000,B10000,B10000,B11000,B11000,
B10000,B10000};
//step 2 of 3
byte stepF1[8] =
{B01111,B01111,B01100,B00111,B00111,B00111,
B00000,B00000};
byte stepF2[8] =
{B00000,B00011,B00000,B00011,B00111,B00011,
B00001,B00000};
byte stepF3[8] =
{B11111,B11111,B11111,B11111,B11111,B11001,
B00000,B00000};
byte stepF4[8] =
{B11111,B11111,B10111,B10111,B11111,B11111,
B11111,B11111};
byte stepF5[8] =
{B11111,B11111,B10111,B11000,B11100,B11110,
B11110,B11100};
byte stepF6[8] =
{B00000,B10000,B10000,B11000,B11000,B11000,
B00000,B11100};
//step 3 of 3
byte stepG1[8] =
{B00001,B00011,B00001,B00000,B00000,B00000,
B00000,B00000};

```

```

byte stepG2[8] =
{B00000,B00000,B00011,B00000,B00011,B00111,
B00011,B00001};
byte stepG3[8] =
{B11111,B11111,B11111,B11111,B11111,B11111,
B01111,B11111};
byte stepG4[8] =
{B00000,B11111,B11111,B10111,B10111,B11111,
B11111,B11111};
byte stepG5[8] =
{B10000,B11000,B11100,B11100,B11100,B11110,
B00110,B00000};
byte stepG6[8] =
{B00000,B00000,B10000,B10000,B11000,B11000,
B11000,B00000};
//setting pixels for man slipping left
//step 1 of 3
byte slipD1[8] =
{B00011,B00011,B00001,B00000,B00000,B00000,
B00000,B00000};
byte slipD2[8] =
{B00000,B00001,B00000,B00011,B00111,B00011,
B00001,B00011};
byte slipD3[8] =
{B11111,B11111,B11111,B11111,B11111,B11111,
B00011,B00001};
byte slipD4[8] =
{B11111,B11111,B10111,B10111,B11111,B11111,
B11111,B11111};
byte slipD5[8] =
{B11100,B11100,B11000,B11000,B10000,B11110,
B11110,B11100};
byte slipD6[8] =
{B00000,B11000,B11100,B11110,B11110,B11100,
B11000,B11100};
//step 2 of 3
byte slipE1[8] =
{B00111,B00111,B00111,B00011,B00011,B00000,
B00000,B00000};
byte slipE2[8] =
{B00000,B00011,B00001,B00111,B01111,B00111,
B00011,B00111};
byte slipE3[8] =
{B11111,B11111,B11111,B11111,B11111,B11111,
B00111,B00011};
byte slipE4[8] =
{B11111,B11111,B01111,B01111,B11111,B11111,
B11111,B11111};
byte slipE5[8] =
{B11000,B11000,B10000,B00000,B10000,B11100,
B11100,B11000};
byte slipE6[8] =
{B00000,B00000,B11000,B11100,B11100,B11000,
B10000,B11000};
//step 3 of 3
byte slipF1[8] =
{B01111,B01111,B00111,B00111,B00011,B00001,
B00000,B00000};
byte slipF2[8] =
{B00001,B00111,B00011,B01111,B11111,B01111,
B00111,B01111};
byte slipF3[8] =
{B11111,B11111,B11111,B11111,B11111,B11111,
B01111,B00111};
byte slipF4[8] =
{B11110,B11111,B11111,B11111,B11111,B11111,
B11111,B11111};

byte slipF5[8] =
{B00000,B00000,B00000,B00000,B00000,B11000,
B11000,B10000};
byte slipF6[8] =
{B00000,B00000,B10000,B11000,B11000,B10000,
B00000,B10000};
// END setting all the pixel-art
// START putting al the pixel-art we defined before
in his place for each animation part
// standing man animation part
void standingManO()
int a = 0;
int b = a + 1;
int c = a + 2;
lcd.createChar(1, stepA1);
lcd.createChar(2, stepA2);
lcd.createChar(3, stepA3);
lcd.createChar(4, stepA4);
lcd.createChar(5, stepA5);
lcd.createChar(6, stepA6);
lcd.setCursor(a,1);
lcd.write(1);
lcd.setCursor(a,0);
lcd.write(2);
lcd.setCursor(b,1);
lcd.write(3);
lcd.setCursor(b,0);
lcd.write(4);
lcd.setCursor(c,1);
lcd.write(5);
lcd.setCursor(c,0);
lcd.write(6);
}
// running man to the right animation part
void runningManR() {
for (int a=0;a<11;a+=4){
int b = a + 1;
int c = a + 2;
int d = a + 3;
int e = a + 4;
int f = a + 5;
int g = a + 6;
lcd.clear();
lcd.createChar(1, stepB1);
lcd.createChar(2, stepB2);
lcd.createChar(3, stepB3);
lcd.createChar(4, stepB4);
lcd.createChar(5, stepB5);
lcd.createChar(6, stepB6);
lcd.setCursor(b,1);
lcd.write(1);
lcd.setCursor(b,0);
lcd.write(2);
lcd.setCursor(c,1);
lcd.write(3);
lcd.setCursor(c,0);
lcd.write(4);
lcd.setCursor(d,1);
lcd.write(5);
lcd.setCursor(d,0);
lcd.write(6);
delay(x);
lcd.clear();
lcd.createChar(1, stepC1);
lcd.createChar(2, stepC2);
lcd.createChar(3, stepC3);
lcd.createChar(4, stepC4);
lcd.createChar(5, stepC5);
lcd.createChar(6, stepC6);
}
}

```

```

lcd.setCursor(c,1);
lcd.write(1);
lcd.setCursor(c,0);
lcd.write(2);
lcd.setCursor(d,1);
lcd.write(3);
lcd.setCursor(d,0);
lcd.write(4);
lcd.setCursor(e,1);
lcd.write(5);
lcd.setCursor(e,0);
lcd.write(6);
delay(x);
lcd.clear();
lcd.createChar(1, stepB1);
lcd.createChar(2, stepB2);
lcd.createChar(3, stepB3);
lcd.createChar(4, stepB4);
lcd.createChar(5, stepB5);
lcd.createChar(6, stepB6);
lcd.setCursor(d,1);
lcd.write(1);
lcd.setCursor(d,0);
lcd.write(2);
lcd.setCursor(e,1);
lcd.write(3);
lcd.setCursor(e,0);
lcd.write(4);
lcd.setCursor(f,1);
lcd.write(5);
lcd.setCursor(f,0);
lcd.write(6);
delay(x);
lcd.clear();
lcd.createChar(1, stepD1);
lcd.createChar(2, stepD2);
lcd.createChar(3, stepD3);
lcd.createChar(4, stepD4);
lcd.createChar(5, stepD5);
lcd.createChar(6, stepD6);
lcd.setCursor(e,1);
lcd.write(1);
lcd.setCursor(e,0);
lcd.write(2);
lcd.setCursor(f,1);
lcd.write(3);
lcd.setCursor(f,0);
lcd.write(4);
lcd.setCursor(g,1);
lcd.write(5);
lcd.setCursor(g,0);
lcd.write(6);
delay(x);
}
// slipping man the the right animation part
void ManslipR(){
int a = 13;
int b = a + 1;
int c = a + 2;
lcd.clear();
lcd.createChar(1, slipA1);
lcd.createChar(2, slipA2);
lcd.createChar(3, slipA3);
lcd.createChar(4, slipA4);
lcd.createChar(5, slipA5);
lcd.createChar(6, slipA6);
lcd.setCursor(a,1);
lcd.write(1);

lcd.setCursor(a,0);
lcd.write(2);
lcd.setCursor(b,1);
lcd.write(3);
lcd.setCursor(b,0);
lcd.write(4);
lcd.setCursor(c,1);
lcd.write(5);
lcd.setCursor(c,0);
lcd.write(6);
delay(x);
lcd.clear();
lcd.createChar(1, slipB1);
lcd.createChar(2, slipB2);
lcd.createChar(3, slipB3);
lcd.createChar(4, slipB4);
lcd.createChar(5, slipB5);
lcd.createChar(6, slipB6);
lcd.setCursor(a,1);
lcd.write(1);
lcd.setCursor(a,0);
lcd.write(2);
lcd.setCursor(b,1);
lcd.write(3);
lcd.setCursor(b,0);
lcd.write(4);
lcd.setCursor(c,1);
lcd.write(5);
lcd.setCursor(c,0);
lcd.write(6);
delay(x);
lcd.createChar(1, slipC1);
lcd.createChar(2, slipC2);
lcd.createChar(3, slipC3);
lcd.createChar(4, slipC4);
lcd.createChar(5, slipC5);
lcd.createChar(6, slipC6);
lcd.setCursor(a,1);
lcd.write(1);
lcd.setCursor(a,0);
lcd.write(2);
lcd.setCursor(b,1);
lcd.write(3);
lcd.setCursor(b,0);
lcd.write(4);
lcd.setCursor(c,1);
lcd.write(5);
lcd.setCursor(c,0);
lcd.write(6);
delay(x);
}
// running man to the left animation part
void runningManL(){
for (int a=16;a>4;a-=4){
int b = a - 1;
int c = a - 2;
int d = a - 3;
int e = a - 4;
int f = a - 5;
int g = a - 6;
lcd.clear();
lcd.createChar(1, stepE1);
lcd.createChar(2, stepE2);
lcd.createChar(3, stepE3);
lcd.createChar(4, stepE4);
lcd.createChar(5, stepE5);
lcd.createChar(6, stepE6);
lcd.setCursor(d,1);
lcd.write(1);

```

```

lcd.setCursor(d,0);
lcd.write(2);
lcd.setCursor(c,1);
lcd.write(3);
lcd.setCursor(c,0);
lcd.write(4);
lcd.setCursor(b,1);
lcd.write(5);
lcd.setCursor(b,0);
lcd.write(6);
delay(x);
lcd.clear();
lcd.createChar(1, stepF1);
lcd.createChar(2, stepF2);
lcd.createChar(3, stepF3);
lcd.createChar(4, stepF4);
lcd.createChar(5, stepF5);
lcd.createChar(6, stepF6);
lcd.setCursor(e,1);
lcd.write(1);
lcd.setCursor(e,0);
lcd.write(2);
lcd.setCursor(d,1);
lcd.write(3);
lcd.setCursor(d,0);
lcd.write(4);
lcd.setCursor(c,1);
lcd.write(5);
lcd.setCursor(c,0);
lcd.write(6);
delay(x);
lcd.clear();
lcd.createChar(1, stepE1);
lcd.createChar(2, stepE2);
lcd.createChar(3, stepE3);
lcd.createChar(4, stepE4);
lcd.createChar(5, stepE5);
lcd.createChar(6, stepE6);
lcd.setCursor(f,1);
lcd.write(1);
lcd.setCursor(f,0);
lcd.write(2);
lcd.setCursor(e,1);
lcd.write(3);
lcd.setCursor(e,0);
lcd.write(4);
lcd.setCursor(d,1);
lcd.write(5);
lcd.setCursor(d,0);
lcd.write(6);
delay(x);
lcd.clear();
lcd.createChar(1, stepG1);
lcd.createChar(2, stepG2);
lcd.createChar(3, stepG3);
lcd.createChar(4, stepG4);
lcd.createChar(5, stepG5);
lcd.createChar(6, stepG6);
lcd.setCursor(g,1);
lcd.write(1);
lcd.setCursor(g,0);
lcd.write(2);
lcd.setCursor(f,1);
lcd.write(3);
lcd.setCursor(f,0);
lcd.write(4);
lcd.setCursor(e,1);
lcd.write(5);
lcd.setCursor(e,0);

lcd.write(6);
delay(x);
}
// slipping man to the left animation part
void ManslipL(){
int a = 1;
int b = a + 1;
int c = a + 2;
lcd.clear();
lcd.createChar(1, slipD1);
lcd.createChar(2, slipD2);
lcd.createChar(3, slipD3);
lcd.createChar(4, slipD4);
lcd.createChar(5, slipD5);
lcd.createChar(6, slipD6);
lcd.setCursor(a,1);
lcd.write(1);
lcd.setCursor(a,0);
lcd.write(2);
lcd.setCursor(b,1);
lcd.write(3);
lcd.setCursor(b,0);
lcd.write(4);
lcd.setCursor(c,1);
lcd.write(5);
lcd.setCursor(c,0);
lcd.write(6);
delay(x);
lcd.clear();
lcd.createChar(1, slipE1);
lcd.createChar(2, slipE2);
lcd.createChar(3, slipE3);
lcd.createChar(4, slipE4);
lcd.createChar(5, slipE5);
lcd.createChar(6, slipE6);
lcd.setCursor(a,1);
lcd.write(1);
lcd.setCursor(a,0);
lcd.write(2);
lcd.setCursor(b,1);
lcd.write(3);
lcd.setCursor(b,0);
lcd.write(4);
lcd.setCursor(c,1);
lcd.write(5);
lcd.setCursor(c,0);
lcd.write(6);
delay(x);
lcd.createChar(1, slipF1);
lcd.createChar(2, slipF2);
lcd.createChar(3, slipF3);
lcd.createChar(4, slipF4);
lcd.createChar(5, slipF5);
lcd.createChar(6, slipF6);
lcd.setCursor(a,1);
lcd.write(1);
lcd.setCursor(a,0);
lcd.write(2);
lcd.setCursor(b,1);
lcd.write(3);
lcd.setCursor(b,0);
lcd.write(4);
lcd.setCursor(c,1);
lcd.write(5);
lcd.setCursor(c,0);
lcd.write(6);
delay(x);
}

```



```

void loading(){
  runningManR();
  //ManslipR();
  //runningManL();
  //ManslipL();
}

```

### 3. Jarak Tempuh Code

```

int point_setting_jarak = 1;//1.limit 2.satuan
void menu_setting_jarak(){
  lcd.setCursor(2,0);
  lcd.print("1. Set Jarak ");
  lcd.setCursor(2,1);
  lcd.print("2. Set Satuan ");
  while(1){
    if (point_setting_jarak ==
1){lcd.setCursor(0,0);lcd.print(">");}
    else if(point_setting_jarak ==
2){ lcd.setCursor(0,1);lcd.print(">");}
    if(digitalRead(DOWN) == LOW){
      buzzer_(1, 75);
      delay(delay_btn);
      point_setting_jarak += 1;
      lcd.setCursor(0,0);lcd.print(" ");
      if(point_setting_jarak >= 2){point_setting_jarak
= 2;}
    }
    if(digitalRead(UP) == LOW){
      buzzer_(1, 75);
      delay(delay_btn);
      point_setting_jarak -= 1;
      lcd.setCursor(0,1);lcd.print(" ");
      if(point_setting_jarak <= 1){point_setting_jarak
= 1;}
    }
    if(digitalRead(OK) == LOW){
      buzzer_(1, 75);
      delay(delay_btn);
      if(digitalRead(OK) == LOW){
        buzzer_(3, 75);
        idx_posisi = 1;
        point_menu = 1;
        lcd.clear();
        break;
      }
    }
    else{
      if (point_setting_jarak == 1){
        lcd.clear();
        set_limit();
      }
      else if(point_setting_jarak == 2){
        lcd.clear();
        set_satuan();
      }
      point_setting_jarak = 1;
      //break;
    }
  }
}
void set_limit(){
  lcd.setCursor(0,0);
  lcd.print("PENGATURAN LIMIT");
  lcd.setCursor(0,1);
  lcd.print(limit_jarak);
  lcd.print(satuan);
  while(1){

```

```

if(temp_limit_jarak != limit_jarak or limit_jarak
== 0){
  lcd.setCursor(0,1);
  lcd.print(" ");
  lcd.setCursor(0,1);
  lcd.print(limit_jarak);
  lcd.print(satuan);
  temp_limit_jarak = limit_jarak;
}
// else{
//   lcd.print(limit_jarak);
//   lcd.print(" ");
//   lcd.print(satuan);
// }
if(digitalRead(UP) == LOW){
  buzzer_(1, 75);
  delay(delay_btn);
  //limit_jarak += 1;
  if(satuan == " Km"){
    limit_jarak += 1000;
  }
  else if(satuan == " M"){
    limit_jarak += 10;
  }
  while(digitalRead(UP) == LOW){
    delay(50);
    //limit_jarak += 1;
    if(satuan == " Km"){
      limit_jarak += 1000;
    }
    else if(satuan == " M"){
      limit_jarak += 10;
    }
  }
  if(temp_limit_jarak != limit_jarak){
    lcd.setCursor(0,1);
    lcd.print(" ");
    lcd.setCursor(0,1);
    lcd.print(limit_jarak);
    lcd.print(satuan);
    temp_limit_jarak = limit_jarak;
  }
}
if(digitalRead(DOWN) == LOW){
  buzzer_(1, 75);
  delay(delay_btn);
  //limit_jarak -= 1;
  if(satuan == " Km"){
    limit_jarak -= 1000;
  }
  else if(satuan == " M"){
    limit_jarak -= 10;
  }
}
if(limit_jarak < 0){
  limit_jarak = 0;
}
while(digitalRead(DOWN) == LOW){
  delay(50);
  //limit_jarak -= 1;
  if(satuan == " Km"){
    limit_jarak -= 1000;
  }
  else if(satuan == " M"){
    limit_jarak -= 10;
  }
}
if(limit_jarak < 0){
  limit_jarak = 0;
}
}

```

```

        if(temp_limit_jarak != limit_jarak or
limit_jarak == 0){
            lcd.setCursor(0,1);
            lcd.print(" ");
            lcd.setCursor(0,1);
            lcd.print(limit_jarak);
            lcd.print(satuan);
            temp_limit_jarak = limit_jarak;
        }
    }

    if(digitalRead(OK) == LOW){
        buzzer_(1, 75);
        delay(delay_btn);
        // if(digitalRead(OK) == LOW){
        //   buzzer_(3, 75);
        //   idx_posisi = 0;
        //   lcd.clear();
        //   break;
        // }
        // else{
        //   //idx_posisi = 0;
        //   lcd.clear();
        //   lcd.setCursor(0,0);
        //   lcd.print("TERSIMPAN...");
        //   delay(2000);
        //   lcd.clear();
        //   lcd.setCursor(2,0);
        //   lcd.print("1. Set Jarak ");
        //   lcd.setCursor(2,1);
        //   lcd.print("2. Set Satuan ");
        //   break;
        // }
    }
}

int point_satuan = 1;
void set_satuan(){
    lcd.setCursor(2,0);
    lcd.print("1. Km");
    lcd.setCursor(2,1);
    lcd.print("2. M");
    while(1){
        if (point_satuan ==
1){lcd.setCursor(0,0);lcd.print(">");}
        else if(point_satuan ==
2){lcd.setCursor(0,1);lcd.print(">");}
        if(digitalRead(DOWN) == LOW){
            buzzer_(1, 75);
            delay(delay_btn);
            point_satuan += 1;
            lcd.setCursor(0,0);lcd.print(" ");
            if(point_satuan >= 2){point_satuan = 2;}
        }
        if(digitalRead(UP) == LOW){
            buzzer_(1, 75);
            delay(delay_btn);
            point_satuan -= 1;
            lcd.setCursor(0,1);lcd.print(" ");
            if(point_satuan <= 1){point_satuan = 1;}
        }
    }
    if(digitalRead(OK) == LOW){
        buzzer_(1, 75);
        delay(delay_btn);
        // if(digitalRead(OK) == LOW){
        //   buzzer_(3, 75);
        //   idx_posisi = 0;

```

```

        // point_menu = 1;
        // lcd.clear();
        // break;
        // }
        // else{
        //   if (point_satuan == 1){
        //     satuan = " Km";
        //   }
        //   else if(point_satuan == 2){
        //     satuan = " M";
        //   }
        //   point_satuan = 1;
        //   //idx_posisi = 0;
        //   lcd.clear();
        //   lcd.setCursor(0,0);
        //   lcd.print("TERSIMPAN...");
        //   delay(2000);
        //   lcd.clear();
        //   lcd.setCursor(2,0);
        //   lcd.print("1. Set Jarak ");
        //   lcd.setCursor(2,1);
        //   lcd.print("2. Set Satuan ");
        //   break;
        // }
    }
}

```

#### 4. Waktu Tempuh Code

```

int point_setting_waktu = 1;//1.limit
2.satuan_waktu
void menu_setting_waktu(){
    lcd.setCursor(2,0);
    lcd.print("1. Set Waktu ");
    lcd.setCursor(2,1);
    lcd.print("2. Set Satuan ");
    while(1){
        if (point_setting_waktu ==
1){lcd.setCursor(0,0);lcd.print(">");}
        else if(point_setting_waktu ==
2){lcd.setCursor(0,1);lcd.print(">");}
        if(digitalRead(DOWN) == LOW){
            buzzer_(1, 75);
            delay(delay_btn);
            point_setting_waktu += 1;
            lcd.setCursor(0,0);lcd.print(" ");
            if(point_setting_waktu >=
2){point_setting_waktu = 2;}
        }
        if(digitalRead(UP) == LOW){
            buzzer_(1, 75);
            delay(delay_btn);
            point_setting_waktu -= 1;
            lcd.setCursor(0,1);lcd.print(" ");
            if(point_setting_waktu <=
1){point_setting_waktu = 1;}
        }
    }
    if(digitalRead(OK) == LOW){
        buzzer_(1, 75);
        delay(delay_btn);
        if(digitalRead(OK) == LOW){
            buzzer_(3, 75);
            idx_posisi = 1;
            point_menu = 1;
            lcd.clear();
            break;
        }
    }
}

```

```

    }
    else{
        if (point_setting_waktu == 1){
            lcd.clear();
            set_limit_waktu();
        }
        else if(point_setting_waktu == 2){
            lcd.clear();
            set_satuan_waktu();
        }
        point_setting_waktu = 1;
        //break;
    }
}
}
}
}
void set_limit_waktu(){
    lcd.setCursor(0,0);
    lcd.print("PENGATURAN WAKTU");
    lcd.setCursor(0,1);
    lcd.print(limit_waktu);
    lcd.print(satuan_waktu);
    while(1){
        if(temp_limit_waktu != limit_waktu or
limit_waktu == 0){
            lcd.setCursor(0,1);
            lcd.print(" ");
            lcd.setCursor(0,1);
            lcd.print(limit_waktu);
            lcd.print(satuan_waktu);
            temp_limit_waktu = limit_waktu;
        }
        // else{
        // lcd.print(limit_waktu);
        // lcd.print(" ");
        // lcd.print(satuan_waktu);
        // }
        if(digitalRead(UP) == LOW){
            buzzer_(1, 75);
            delay(delay_btn);
            limit_waktu += 1;
            while(digitalRead(UP) == LOW){
                delay(50);
                limit_waktu += 1;
                if(temp_limit_waktu != limit_waktu){
                    lcd.setCursor(0,1);
                    lcd.print(" ");
                    lcd.setCursor(0,1);
                    lcd.print(limit_waktu);
                    lcd.print(satuan_waktu);
                    temp_limit_waktu = limit_waktu;
                }
            }
        }
        if(digitalRead(DOWN) == LOW){
            buzzer_(1, 75);
            delay(delay_btn);
            limit_waktu -= 1;
            if(limit_waktu < 0){
                limit_waktu = 0;
            }
            while(digitalRead(DOWN) == LOW){
                delay(50);
                limit_waktu -= 1;
                if(limit_waktu < 0){
                    limit_waktu = 0;
                }
                if(temp_limit_waktu != limit_waktu or
limit_waktu == 0){
                    lcd.setCursor(0,1);
                    lcd.print(" ");
                    lcd.setCursor(0,1);
                    lcd.print(limit_waktu);
                    lcd.print(satuan_waktu);
                    temp_limit_waktu = limit_waktu;
                }
            }
        }
        if(digitalRead(OK) == LOW){
            buzzer_(1, 75);
            delay(delay_btn);
            // if(digitalRead(OK) == LOW){
            // buzzer_(3, 75);
            // idx_posisi = 0;
            // point_menu = 1;
            // lcd.clear();
        }
    }
}
int point_satuan_waktu = 1;
void set_satuan_waktu(){
    lcd.setCursor(2,0);
    lcd.print("1. Bulan");
    lcd.setCursor(2,1);
    lcd.print("2. Detik");
    while(1){
        if (point_satuan_waktu ==
1){lcd.setCursor(0,0);lcd.print(">");}
        else if(point_satuan_waktu ==
2){lcd.setCursor(0,1);lcd.print(">");}
        if(digitalRead(DOWN) == LOW){
            buzzer_(1, 75);
            delay(delay_btn);
            point_satuan_waktu += 1;
            lcd.setCursor(0,0);lcd.print(" ");
            if(point_satuan_waktu >=
2){point_satuan_waktu = 2;}
        }
        if(digitalRead(UP) == LOW){
            buzzer_(1, 75);
            delay(delay_btn);
            point_satuan_waktu -= 1;
            lcd.setCursor(0,1);lcd.print(" ");
            if(point_satuan_waktu <=
1){point_satuan_waktu = 1;}
        }
        if(digitalRead(OK) == LOW){
            buzzer_(1, 75);
            delay(delay_btn);
            // if(digitalRead(OK) == LOW){
            // buzzer_(3, 75);
            // idx_posisi = 0;
            // point_menu = 1;
            // lcd.clear();
        }
    }
}

```

```

// break;
// }
// else{
if (point_satuan_waktu == 1){
    satuan_waktu = " Bulan";
}
else if(point_satuan_waktu == 2){
    satuan_waktu = " Detik";
}
point_satuan_waktu = 1;
idx_posisi = 0;
lcd.clear();
lcd.setCursor(0,0);
lcd.print("TERSIMPAN...");
delay(2000);
lcd.clear();
lcd.setCursor(2,0);
lcd.print("1. Set Waktu ");
lcd.setCursor(2,1);
lcd.print("2. Set Satuan ");
break;
// }
}
}
}

```

## 5. Menu File Code

```

void menu_depan(){
    lcd.setCursor(2,0);
    lcd.print("1. Setting ");
    lcd.setCursor(2,1);
    lcd.print("2. Mulai ");
    while(1){
        if (point_menu ==
1){lcd.setCursor(0,0);lcd.print("->");}
        else if(point_menu ==
2){lcd.setCursor(0,1);lcd.print("->");}
        if(digitalRead(DOWN) == LOW){
            buzzer_(1, 75);
            delay(delay_btn);
            point_menu += 1;
            lcd.setCursor(0,0);lcd.print(" ");
            if(point_menu >= 2){point_menu = 2;}
        }
        if(digitalRead(UP) == LOW){
            buzzer_(1, 75);
            delay(delay_btn);
            point_menu -= 1;
            lcd.setCursor(0,1);lcd.print(" ");
            if(point_menu <= 1){point_menu = 1;}
        }
        if(digitalRead(OK) == LOW){
            buzzer_(1, 75);
            delay(delay_btn);
            if (point_menu == 1){idx_posisi = 1;}
            else if(point_menu == 2){idx_posisi = 2;}
            point_menu = 1;
            lcd.clear();

```

```

        break;
    }
}
int point_setting = 1;//1.jarak tempuh 2.waktu tempuh
void menu_setting(){
    lcd.setCursor(2,0);
    lcd.print("1.Jarak Tempuh");
    lcd.setCursor(2,1);
    lcd.print("2.Waktu Tempuh");
    while(1){
        if (point_setting ==
1){lcd.setCursor(0,0);lcd.print("->");}
        else if(point_setting ==
2){lcd.setCursor(0,1);lcd.print("->");}
        if(digitalRead(DOWN) == LOW){
            buzzer_(1, 75);
            delay(delay_btn);
            point_setting += 1;
            lcd.setCursor(0,0);lcd.print(" ");
            if(point_setting >= 2){point_setting = 2;}
        }
        if(digitalRead(UP) == LOW){
            buzzer_(1, 75);
            delay(delay_btn);
            point_setting -= 1;
            lcd.setCursor(0,1);lcd.print(" ");
            if(point_setting <= 1){point_setting = 1;}
        }
        if(digitalRead(OK) == LOW){
            buzzer_(1, 75);
            delay(delay_btn);
            if(digitalRead(OK) == LOW){
                buzzer_(3, 75);
                idx_posisi = 0;
                point_menu = 1;
                lcd.clear();
                break;
            }
            else{
                if (point_setting == 1){
                    lcd.clear();
                    menu_setting_jarak();
                }
                else if(point_setting == 2){
                    lcd.clear();
                    menu_setting_waktu();
                }
                point_setting = 1;
                break;
            }
        }
    }
}
}
}

```

**Lampiran B. Dokumentasi Penelitian.**



