



REPUBLIK INDONESIA
KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA

SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan : EC00202272478, 6 Oktober 2022

Pencipta

Nama : **Sugeng, Asniar Aliyu S.T., M.Eng. dkk**
Alamat : Dukuhan Tanjung RT 04/02, Desa Tanjunganom, Gabus, Pati, Jawa Tengah, Pati, JAWA TENGAH, 59173
Kewarganegaraan : Indonesia

Pemegang Hak Cipta

Nama : **LPPMI ITNY**
Alamat : Jalan Babarsari, Caturtunggal, Depok, Sleman, Yogyakarta, Yogyakarta, DI YOGYAKARTA, 55281
Kewarganegaraan : Indonesia

Jenis Ciptaan : **Program Komputer**
Judul Ciptaan : **PROGRAM MIKROKONTROLER UNTUK DESAIN SMARTPHONE DAMAGE ANALYZER**

Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia : 19 Agustus 2022, di Yogyakarta

Jangka waktu perlindungan : Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.

Nomor pencatatan : 000388219

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.
Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.



a.n Menteri Hukum dan Hak Asasi Manusia
Direktur Jenderal Kekayaan Intelektual
u.b.
Direktur Hak Cipta dan Desain Industri

Anggoro Dasananto
NIP.196412081991031002

Disclaimer:

Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan.

LAMPIRAN PENCIPTA

No	Nama	Alamat
1	Sugeng	Dukuh Tanjung RT 04/02, Desa Tanjunganom, Gabus, Pati, Jawa Tengah
2	Asniar Aliyu S.T., M.Eng.	Pogung Kidul RT 01/49, Sinduadi, Mlati, Sleman, Yogyakarta
3	Sudiana S.T., M. Kom	Jl Babarsari, Komplek Yadara Blok II/ No.04, Caturtunggal, Depok, Sleman, Yogyakarta





DOKUMEN

HAK KEKAYAAN INTELEKTUAL KARYA CIPTA

**PROGRAM MIKROKONTROLER UNTUK DESAIN
*SMARTPHONE DAMAGE ANALYZER***

Pencipta :

Sugeng
Asniar Aliyu, ST. M.Eng
Sudiana, ST. M.Kom

INSTITUT TEKNOLOGI NASIONAL YOGYAKARTA

2022

DESAIN SMARTPHONE DAMAGE ANALYZER

Perancangan Menjadi teknisi smartphone harus memiliki beberapa alat servis elektronika untuk membantu dalam pekerjaan, seperti Multimeter Digital, MBR (*Micro Buck Regulator*), *Power Supply*, dan lain lain. Alat tersebut berperan penting dalam mengetahui keadaan komponen itu dalam keadaan baik atau tidak. Multimeter digunakan sebagai alat untuk memperbaiki atau membuat sebuah rangkaian listrik. Alat ini berfungsi untuk mendeteksi langsung kondisi dari komponen listrik seperti barang-barang elektronik. Multimeter ini akan dapat mengetahui ukuran komponennya contohnya resistor, kapasitor, lilitan, dan transistor serta kondisinya. Karena kebanyakan komponen SMD sekarang memiliki warna yang sama sulit untuk dibedakan antara itu jenis kapasitor, resistor ataupun lilitan, sedangkan ada alat lainya seperti MBR ini digunakan sebagai penghancur komponen hubung-singkat (*short circuit*) memiliki arus dan tegangan tinggi yang bisa diatur. Catu daya (*power supply*) berfungsi untuk menganalisa sebuah kerusakan pada *smartphone* atau bisa sebagai pengganti baterai. Dalam menentukan jenis kerusakan pada *smartphone* yang sekarang masih menggunakan alat ukur independen untuk melakukan pengukuran seperti arus, tegangan, kondisi komponen SMD, dan lain lain. Berdasarkan hal tersebut, penulis mencoba mendesain *Smartphone Damage Analyzer* di mana alat dilengkapi dengan Multimeter Digital, MBR, *Power Supply*, dalam satu modul yang dikontrol menggunakan Arduino Mega2560

1. Prinsip Kerja

Prinsip kerja sistem ini berdasarkan penggunaanya adalah:

a) Pembacaan arus dari *smartphone*

Dari *power supply smps* 12V 5A akan terhubung dengan modul *step down* 2A 12V dan ACS712 sebagai pembaca arus, di modul *step down* yang terdapat pontensiometer untuk mengatur berapa tegangan yang diinginkan. Contoh: potensio diatur sampai di LCD menampilkan tegangan

3.7V, kemudian hubungkan kabel negatif alat ke konektor negatif pada mesin *smartphone*. Selanjutnya hubungkan kabel positif alat ke konektor positif pada mesin *smartphone* tekan tombol *power* pada mesin *smartphone* maka data hasil pembacaan arus pada mesin *smartphone* akan tampil di LCD.

b) Pembacaan Komponen

Pada pembacaan komponen ada untuk mengukur hambatan, mengukur tegangan dan mengukur dioda (dioda dan kontinuitas/ untuk mengetahui jalur terputus atau tidak). Untuk pembacaan hambatan, tegangan dan dioda (dioda dan kontinuitas) ini akan terpisah tidak jadi menjadi satu, untuk memilihnya mana yang akan digunakan hanya menekan saklar *toggle*.

c) Penggunaan MBR

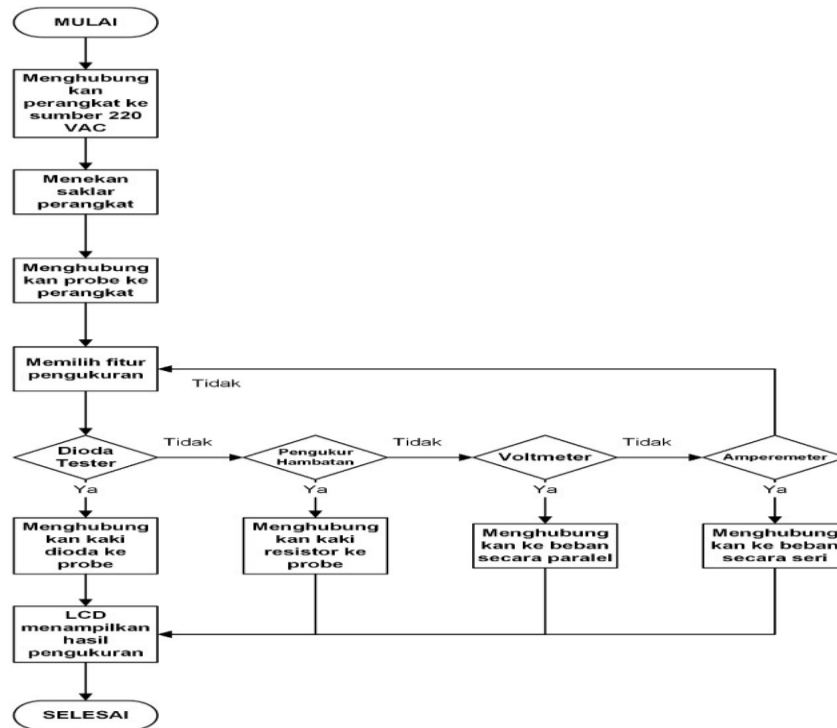
Dari *power supply SMPS 12V 5A* akan terhubung dengan modul *step down 8A* dan ACS712. Pada *step down* akan ada potensiometer untuk mengatur tegangan dan arus serta luarannya akan ditambahkan rangkaian *short protection* untuk mencegah terbaliknya pemasangan kutub positif dan negatif. Saat terjadi salah pemasangan maka tegangan yang keluar akan terputus secara otomatis.

d) Penggunaan QC3.0

QC3.0 yang mendapatkan tegangan dari *step down 12V 2A* terhubung dengan sensor arus ACS712, pada saat USB *smartphone* terhubung maka akan ada proses pembacaan arus dari ACS712 lalu akan di proses ke Arduino Mega2560 berupa data pembacaan yang akan ditampilkan di LCD 20X4

2. Algoritma Pemrograman

Perancangan Desain *Smartphone Damage Analyzer* ini memiliki algoritma pemrograman yang ditunjukkan dengan diagram alir pada gambar berikut.



Gambar 1 Diagram alir algoritma pemrograman

3. Listing Program Desain *Smartphone Damage Analyzer*

```

//A1 DAN A2 PSU
//A3 TEGANGAN CHAGER DAN A4 AMPERE CHAGER
//A0 PENGUKUR DC TEGANGN
//A10 BUZZER//D4 BUZZER
//Ohm a11 //8,9,10,11,12,13
//MBR A9 DAN 8
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,20,4);
//MBR
int ac, bc, cc;
long voltc, currentc;
//ohm
#define CH0 12
#define CH1 11
#define CH2 10
#define CH3 9
#define CH4 8
  
```

```

// variables
byte ch_number;
uint32_t res;
const uint32_t res_table[5] = {100, 1000, 10000, 100000, 2000000};
char _buffer[11];
//dioda//buzzer//
int B=4;
float D,Di=0;
////BATAS////
///CHAGER///
int aa,ba,ca;
long currenta,powera,aha;
// the 8 arrays that form each segment of the custom numbers
unsigned long msecA = 0;
int sampleA = 0;
float totalChargeA = 0.0;
float averageAmpsA = 0.0;
float ampSecondsA = 0.0;
long ampHoursA = 0;
int analogInput = A3;
float vout = 0.0;
float vin = 0.0;
float R1 = 100000.0; // resistance of R1 (100K) -see text!
float R2 = 10000.0; // resistance of R2 (10K) - see text!
int value = 0;
////BATAS////
///PSU///
int a,b,c;
long volt,current,power,ah;
// the 8 arrays that form each segment of the custom numbers
unsigned long msec = 0;
float time = 0.0;
int sample = 0;
float totalCharge = 0.0;
float averageAmps = 0.0;
float ampSeconds = 0.0;
long ampHours = 0;
////BATAS////
///TEGANGAN///
int analogInputDC = A0;
float voutDC = 0.0;
float vinDC = 0.0;
float R1DC = 100000.0; // resistance of R1 (100K) -see text!
float R2DC = 10000.0;
void setup(){
  Serial.begin(115200);
  lcd.init();

```

```
lcd.backlight();
pinMode(analogInput, INPUT);
pinMode(analogInputDC, INPUT);
pinMode(B,OUTPUT);
```

```
pinMode(CH0, OUTPUT);
pinMode(CH1, OUTPUT);
pinMode(CH2, OUTPUT);
pinMode(CH3, OUTPUT);
pinMode(CH4, OUTPUT);
ch_number = 4;
ch_select(ch_number);
```

```
//batas//
```

```
}
```

```
void loop()
```

```
{
```

```
//MBR
```

```
for(int n=0;n<200;n++){
```

```
cc = analogRead(A9);
```

```
ac = analogRead(A8);
```

```
voltc=voltc+ac;
```

```
currentc=currentc+cc;
```

```
delay(1);
```

```
}
```

```
currentc=(currentc/200-514);
```

```
if(currentc<1)currentc=0;
```

```
currentc=currentc*15;
```

```
voltc=voltc/30;
```

```
Serial.print(voltc);
```

```
Serial.print(" ");
```

```
Serial.println(currentc);
```

```
lcd.setCursor(3,3);
```

```
lcd.print("A");
```

```
bc=(currentc/10)%10;
```

```
lcd.setCursor(2,3);
```

```
lcd.print(bc);
```

```
lcd.setCursor(1,3);
```

```
lcd.print(".");
```

```
bc=(currentc/100)%10;
```

```
lcd.setCursor(0,3);
```

```
lcd.print(bc);
```

```
lcd.setCursor(0,3);
```

```
lcd.print("MBR:");
```

```
lcd.setCursor(9,3);
```

```
lcd.print("V");
```

```
bc=voltc%10;
```



```

lcd.setCursor(8,3);
lcd.print(bc);
bc=(voltc/10)%10;
lcd.setCursor(7,3);
lcd.print(bc);
lcd.setCursor(6,3);
lcd.print(".");
bc=(voltc/100)%10;
lcd.setCursor(5,3);
lcd.print(bc);
bc=(voltc/1000)%10;
lcd.setCursor(4,3);
if(volt>999)lcd.print(bc);
else lcd.print(" ");
//BATAS
////CHAGER//
for(int j=0;j<200;j++){
ca = analogRead(A0);
currenta=currenta+ca;
delay(1);
}
currenta=(currenta/200-514);
if(currenta<1)currenta=0;
currenta=currenta*15;
//-----
samplea = samplea + 1;
mseca = millis();
time = (float) mseca / 1000.0;
totalChargea = totalChargea + (currenta);
averageAmpsA = totalChargea / samplea;
ampSecondsA = averageAmpsA*time;
ampHoursA = ampSecondsA/3600;
Serial.print(currenta);
Serial.print(" ");
Serial.print(powera);
Serial.print(" ");
Serial.print(ampHoursA);
Serial.print(" ");
Serial.println(time);
lcd.setCursor(9,1);
lcd.print("A");
ba=currenta%10;
lcd.setCursor(8,1);
lcd.print(ba);
ba=(currenta/10)%10;
lcd.setCursor(7,1);
lcd.print(ba);

```

```

    lcd.setCursor(6,1);
    lcd.print(".");
    ba=(currenta/100)% 10;
    lcd.setCursor(5,1);
    lcd.print(ba);
    /////BATAS/////
    //chagerV///
    // read the value at analog input
    value = analogRead(analogInput);
    vout = (value * 5.0) / 1024.0; // see text
    vin = vout / (R2/(R1+R2));
    if (vin<0.09) {
    vin=0.0;//statement to quash undesired reading !
    }
    lcd.setCursor(0, 1);
    lcd.print("RGER:");
    lcd.setCursor(0, 0);
    lcd.print("CHA ");
    lcd.print(vin);
    delay(500);
    lcd.setCursor(9, 0);
    lcd.print("V");
    //////////BATAS////////
    ///PSU///
    for(int i=0;i<200;i++){
    c = analogRead(A2);
    a = analogRead(A1);
    volt=volt+a;
    current=current+c;
    delay(1);
    }
    current=(current/200-514);
    if(current<1)current=0;
    current=current*15;
    volt=volt/30;
    power=(volt*current)/1000;
    //-----
    sample = sample + 1;
    msec = millis();
    time = (float) msec / 1000.0;
    totalCharge = totalCharge + (current);
    averageAmps = totalCharge / sample;
    ampSeconds = averageAmps*time;
    ampHours = ampSeconds/3600;

    //-----
    Serial.print(volt);

```

```

Serial.print(" ");
Serial.print(current);
Serial.print(" ");
Serial.print(power);
Serial.print(" ");
Serial.print(ampHours);
Serial.print(" ");
Serial.println(time);
lcd.setCursor(19,0);
lcd.print("V");
b=volt%10;
lcd.setCursor(18,0);
lcd.print(b);
b=(volt/10)%10;
lcd.setCursor(17,0);
lcd.print(b);
lcd.setCursor(16,0);
lcd.print(".");
b=(volt/100)%10;
lcd.setCursor(15,0);
lcd.print(b);
b=(volt/1000)%10;
lcd.setCursor(14,0);
if(volt>999)lcd.print(b);
else lcd.print(" ");
lcd.setCursor(11,0);
lcd.print(" ");
lcd.setCursor(19,1);
lcd.print("A");
b=current%10;
  lcd.setCursor(18,1);
lcd.print(b);
  b=(current/10)%10;
  lcd.setCursor(17,1);
lcd.print(b);
  lcd.setCursor(16,1);
lcd.print(".");
  b=(current/100)%10;
  lcd.setCursor(15,1);
lcd.print(b);
lcd.setCursor(11 ,1);
lcd.print("PSU:"); //unit for the current to be measured
////////BATAS //////////
//tegangan//
// read the value at analog input
value = analogRead(analogInputDC);
voutDC = (value * 5.0) / 1024.0; // see text

```

```

    vinDC = voutDC / (R2/(R1+R2));
    if (vinDC<0.09) {
        vinDC=0.0;//statement to quash undesired reading !
    }
    lcd.setCursor(9, 2);
    lcd.print("V");
    lcd.setCursor(0, 2);
    lcd.print("DC: ");
    lcd.print(vinDC);
    delay(500);
    lcd.setCursor(10,0);
    lcd.print("|");
    lcd.setCursor(10,1);
    lcd.print("|");
    lcd.setCursor(10,2);
    lcd.print("|");
    lcd.setCursor(10,3);
    lcd.print("|");
    ///DIODA///BUZZER//
    D=analogRead(A10);
    Di = D* (5.0/1023.0);
    if(Di>0.08)
    {
        lcd.setCursor(11,2);
        lcd.print("VDID:");
        lcd.setCursor(16,2);
        lcd.print(Di);
        delay(200);
    }
    else {
        byte buzzer [8]=
        {
            0b00001,
            0b00011,
            0b01111,
            0b01111,
            0b01111,
            0b00011,
            0b00001,
            0b00000
        };
        byte sound [8]=
        {
            0b00000,
            0b00100,
            0b00010,
            0b10010,

```

```

0b10010,
0b00010,
0b00100,
0b00000
};
lcd.createChar(1,buzzer);
lcd.createChar(2,sound);
lcd.clear();
lcd.setCursor(11,2);
lcd.print("CONNECT");
lcd.write(byte(1));
lcd.write(byte(2));
if(D<0.02) {
    digitalWrite(B,HIGH);
    delay(500);
    digitalWrite(B,LOW);
}
else {
    digitalWrite(B,LOW);
}
}
}
///ohm
uint16_t volt_image = analogRead(A11) + 1;

if(volt_image >= 550 && ch_number < 4) {
    ch_number++;
    ch_select(ch_number);
    delay(50);
    return;
}
if(volt_image <= 90 && ch_number > 0) {
    ch_number--;
    ch_select(ch_number);
    delay(50);
    return;
}
if(volt_image < 900) {
    float value = (float)volt_image*res/(1023 - volt_image);
    if(value < 1000.0)
        sprintf(_buffer, "%03u.%1u  ", (uint16_t)value, (uint16_t)(value*10)%10);
    else if(value < 10000.0)
        sprintf(_buffer, "%1u.%03u k", (uint16_t)(value/1000), (uint16_t)value%1000);
    else if(value < 100000.0)
        sprintf(_buffer, "%02u.%02u k", (uint16_t)(value/1000), (uint16_t)(value/10)%100);
    else if(value < 1000000.0)
        sprintf(_buffer, "%03u.%1u k", (uint16_t)(value/1000), (uint16_t)(value/100)%10);
    else

```

```

    sprintf(_buffer, "%1u.%03u M", (uint16_t)(value/1000000), (uint16_t)(value/1000)%1000);
    lcd.setCursor(19,3);
    lcd.print((char)244);
}
else
sprintf(_buffer, "--ZERO!--");
lcd.setCursor(11, 3); // move cursor to position (0, 1)
lcd.print(_buffer);
Serial.println(_buffer);
Serial.println();
delay(500); // wait some time
}
void ch_select(byte n) {
switch(n) {
case 0:
    digitalWrite(CH0, LOW);
    digitalWrite(CH1, HIGH);
    digitalWrite(CH2, HIGH);
    digitalWrite(CH3, HIGH);
    digitalWrite(CH4, HIGH);
    break;
case 1:
    digitalWrite(CH0, HIGH);
    digitalWrite(CH1, LOW);
    digitalWrite(CH2, HIGH);
    digitalWrite(CH3, HIGH);
    digitalWrite(CH4, HIGH);
    break;
case 2:
    digitalWrite(CH0, HIGH);
    digitalWrite(CH1, HIGH);
    digitalWrite(CH2, LOW);
    digitalWrite(CH3, HIGH);
    digitalWrite(CH4, HIGH);
    break;
case 3:
    digitalWrite(CH0, HIGH);
    digitalWrite(CH1, HIGH);
    digitalWrite(CH2, HIGH);
    digitalWrite(CH3, LOW);
    digitalWrite(CH4, HIGH);
    break;
case 4:
    digitalWrite(CH0, HIGH);
    digitalWrite(CH1, HIGH);
    digitalWrite(CH2, HIGH);
    digitalWrite(CH3, HIGH);

```



```
    digitalWrite(CH4, LOW);  
  }  
  res = res_table[n];  
}
```