SCRIPT PROGRAM :

```
#include "EmonLib.h"                    // Include Emon Library
EnergyMonitor emon1;                     // Create an instance

#include <Sim800l.h>
#include <SoftwareSerial.h> //is necesary for the library!! TX=11, RX=10
Sim800l Sim800l;  //to declare the library
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x3F,2,1,0,4,5,6,7,3,POSITIVE);

const int analogInPinVolt = A1;  // inisialisasi pin A1 sebagai pembacaan
sensor
const int analogInPinAmp = A0;  // inisialisasi pin A0 sebagai pembacaan
sensor

#define limit_switch 13

/////////ac
float sensor_Value = 0;        // setting nilai default pembacaan sensor 0
float voltageOut = 0;
float voltage = 0;

unsigned int start_times[300];
unsigned int stop_times[300];
unsigned int values[300];

int z,k,m,s,i,zk,zm,zs,n;
int value[300];
int a = 0;
int zero = 0;
/////////ac

float amplitudo_current;
float effective_current;

int tegangan = 127;
double konstanta = 0.005;
int sensor = 0;

int arus, pilih, x;
double current;

void setup() {
  // put your setup code here, to run once:
lcd.begin(16, 2);
Serial.begin(9600);
emon1.current(A0, 111.1); //pin , kalibrasi
Sim800l.begin(); // initializate the library.
//send sms
//Sim800l.sendSms("+540111111111","the text go here")
pinMode(limit_switch, INPUT_PULLUP); //limit switch
digitalWrite(limit_switch, HIGH);

lcd.setCursor(0,0);        // go to the next line
lcd.print("     SKRIPSI    ");
lcd.setCursor(0,1);        // go to the next line
lcd.print(" TEKNIK ELEKTRO ");
delay(1000);
lcd.clear();
}

void loop() {
  // put your main code here, to run repeatedly:
/*
Serial.println("stanby");
if(Serial.available() > 0) {
  if(Serial.read() == 'a') {
     Serial.println("kirim");
     Sim800l.sendSms("08993543075","BISMILLAH");
  }
  delay(500);
}
  x = digitalRead(limit_switch);
Serial.println(x);

Serial.println("stanby");
if(Serial.available() > 0) {
  if(Serial.read() == 'a') {
     Serial.println("kirim");
     Sim800l.sendSms("08993543075","BISMILLAH");
  }
  delay(500);
}
*/
jalan();
//sens_arus();
//sens_tegangan();
//sensor_arus();

}
```

```
void jalan() {
 switch(pilih){
   case 0:{
     Serial.println("case 0");
     lcd.setCursor(15,0);       // go to the next line
     lcd.print("0");
     sens_tegangan();
     sens_arus();
     if(x==0 && digitalRead(limit_switch) == HIGH && voltage > 0 &&
current > 0.10)
     {
      lcd.clear();
       x=1; //tanda ditutup
       pilih=1;
       break;
     }
     break;
   }

   case 1:{
     Serial.println("case 1");
     lcd.setCursor(15,0);       // go to the next line
     lcd.print("1");
     sens_tegangan();
     sens_arus();
     if(x==1 && digitalRead(limit_switch) == LOW && voltage == 0 &&
current < 0.15) {
       lcd.clear();
       x=0;
       pilih=2;
       break;
     }
     /*
     if(voltage >= 100 && current < 0.3) {
       pilih=3;
       break;
     }
     */
     break;
   }

   case 2:{
     lcd.setCursor(15,0);       // go to the next line
     lcd.print("2");
     lcd.setCursor(0,0);       // go to the next line
     lcd.print("   BOX DIBUKA   ");
```

```
    Serial.println("BOX DIBUKA");
    Sim800l.sendSms("085649011848","BOX DIBUKA");
    delay(3000);
    lcd.clear();
    pilih=0;
    break;
  }
  /*
  case 3:{
    Serial.println("ALAT ERROR!!!");
    Sim800l.sendSms("08993543075","ALAT ERROR!!!");
    //delay(3000);
    pilih=4;
    break;
  }

  case 4:{
    Serial.println("ERROR!!!");
    sens_tegangan();
    sens_arus();
    if(voltage<50){
      pilih=0;
      break;
    }
    break;
  }*/
 }//switch
}

void sens_tegangan() {
  read_adc();
  voltage = ((n-524)*2.43)-104;
  if(voltage < 0) {
   voltage=0;
  }
  lcd.setCursor(0,0);        // go to the next line
  lcd.print("TEGANGAN=");
  lcd.setCursor(9,0);        // go to the next line
  lcd.print(voltage);
  Serial.print(voltage,0);
  Serial.print("       ");
}

void sens_arus() {
  current = emon1.calcIrms(1480);  // Calculate Irms only
  Serial.println(current);                   // Irms
  lcd.setCursor(0,1);        // go to the next line
  lcd.print("ARUS (A)=");
  lcd.setCursor(9,1);        // go to the next line
  lcd.print(current);
}

/*
void sens_arus() {
  int sensor_max;
  sensor_max = getMaxValue();

  amplitudo_current = (float)sensor_max/1024*5/200*1000000;
  effective_current = amplitudo_current/1.414;
  //Serial.print("aplitude = ");
  //Serial.print(amplitudo_current,1);
  //Serial.print("    ");
  //Serial.print("effective = ");
  Serial.print(effective_current,1);
  Serial.print("    ");
  arus = effective_current/100;
  Serial.println(arus,1);


  if(effective_current > 500){
  arus = effective_current/1000-0.5;
  }
  else{
  arus = effective_current/1000;
  }
}


int getMaxValue()
{
    int sensorValue;    //value read from the sensor
    int sensorMax = 0;
    uint32_t start_time = millis();
    //Serial.print("start_time = ");
    //Serial.println(start_time);

    while((millis()-start_time) < 1000) //sample for 1000ms
    {
        sensorValue = analogRead(analogInPinAmp);
        if (sensorValue > sensorMax)
        {
            //record the maximum sensor value
            sensorMax = sensorValue;
        }
    }
    return sensorMax;
}
*/

void read_adc() {
  for (int i=0; i<300; i++) {
    value[i] = analogRead(1);

    if(value[i] >= z) {
     z = value[i];
    }
  }
  for (int k=0; k<300; k++) {
    value[k] = analogRead(1);

    if(value[k] >= zk) {
     zk = value[k];
    }
  }
  for (int m=0; m<300; m++) {
    value[m] = analogRead(1);

    if(value[m] >= zm) {
     zm = value[m];
    }
  }
  for (int s=0; s<300; s++) {
    value[s] = analogRead(1);

    if(value[s] >= zs) {
     zs = value[s];
    }
  }
  if((z>=zk)&&(z>=zm)&&(z>=zs)) {
     n=z;
  }
  if((zk>=z)&&(zk>=zm)&&(zk>=zs)) {
     n=zk;
  }
  if((zm>=z)&&(zm>=zk)&&(zm>=zs)) {
     n=zm;
  }
  if((zs>=z)&&(zs>=zm)&&(zs>=z)) {
     n=zs;
  }
  //lcd.setCursor(0,0);
  //lcd.print("ADC = ");
  //lcd.print(n);

  delay(100);
  z=0;
  zk=0;
  zm=0;
  zs=0;
}// void read_ac
```