

LAMPIRAN

1. LISTING PROGRAM

```
#include <delay.h>
#include <stdlib.h>
#include <stdio.h>
#include <mega8.h>
#define on PIND.7
#define pintu PIND.6
#define pir PIND.5
#define kuning PORTB.0
#define merah PORTB.1
#define bunyi PORTB.2
#define balik PORTB.3

int detik;

// Timer1 overflow interrupt service routine
interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
// Reinitialize Timer1 value
TCNT1H=0xD23A >> 8;
TCNT1L=0xD23A & 0xff;
detik++;
}

void trap()
{ while(1){
delay_ms(1);
balik=1; detik=0; }
}

void sensor_pir()
{detik=0;
```

```
while (1)
{
if (detik<=9999999) {
delay_ms (1) ;
bunyi=1;
merah=1;
delay_ms (2000) ;
merah=0;
delay_ms (2000) ;
}
if (detik>=120) {
trap () ;
}
}}

void sensor_on ()
{detik=0;
while (1)
{delay_ms (1) ;
if (detik<=999999999) {
kuning=0;
}delay_ms (1) ;
if (pir==0) {delay_ms (1) ;
sensor_pir () ;
}
if (detik>=2) {delay_ms (1) ;
trap () ;
}}}
```

```

void sensor_pintu()
{detik=0;
while(1)
{delay_ms(1);
if(detik<=999999){
kuning=1;
}delay_ms(1);
if(pir==0){delay_ms(1);
sensor_pir();
}
if(detik>=20){
trap();
}
}}

// Declare your global variables here

void main(void)
{
// Declare your local variables here

// Input/Output Ports initialization
// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=Out Func2=Out
Func1=Out Func0=Out
// State7=T State6=T State5=T State4=T State3=0 State2=0
State1=0 State0=0

PORTB=0x00;
DDRB=0x0F;

```

```

// Port C initialization
// Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTC=0x00;
DDRC=0x00;

// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=P State6=P State5=P State4=P State3=T State2=T State1=T
State0=T
PORTD=0xF0;
DDRD=0x00;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
TCCR0=0x00;
TCNT0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 11.719 kHz
// Mode: Normal top=0xFFFF
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: On
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off

```

```
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x05;
TCNT1H=0xD2;
TCNT1L=0x3A;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer2 Stopped
// Mode: Normal top=0xFF
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
MCUCR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x04;

// USART initialization
```

```

// USART disabled
UCSRB=0x00;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// ADC initialization
// ADC disabled
ADCSRA=0x00;

// SPI initialization
// SPI disabled
SPCR=0x00;

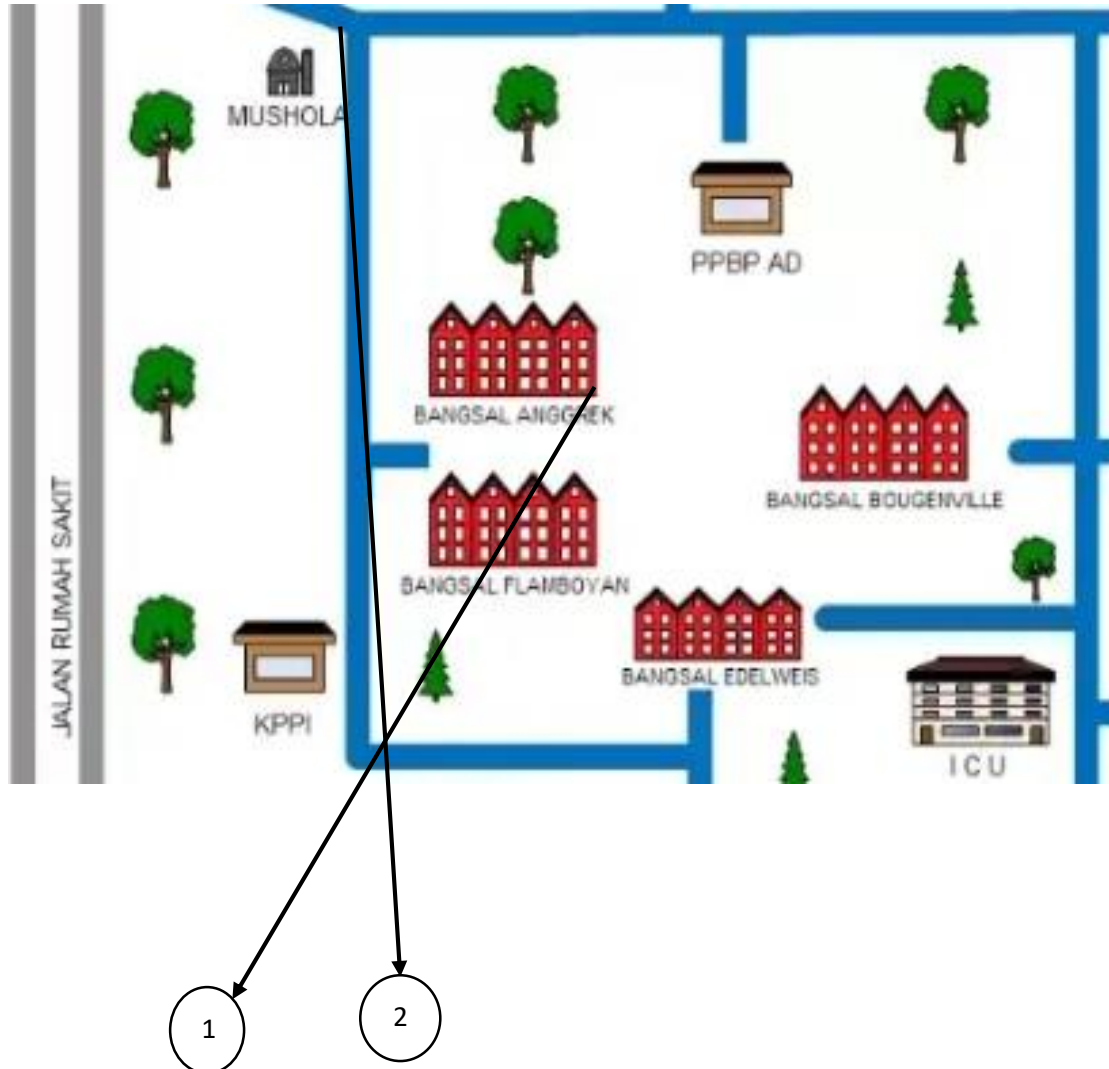
// TWI initialization
// TWI disabled
TWCR=0x00;

// Global enable interrupts
#asm("sei")

while (1)
{
    if(pir==0) sensor_pir();
    if(on==0) sensor_on();
    if(pintu==1) sensor_pintu();
}
}

```

2. GAMBAR LOKASI SENSOR

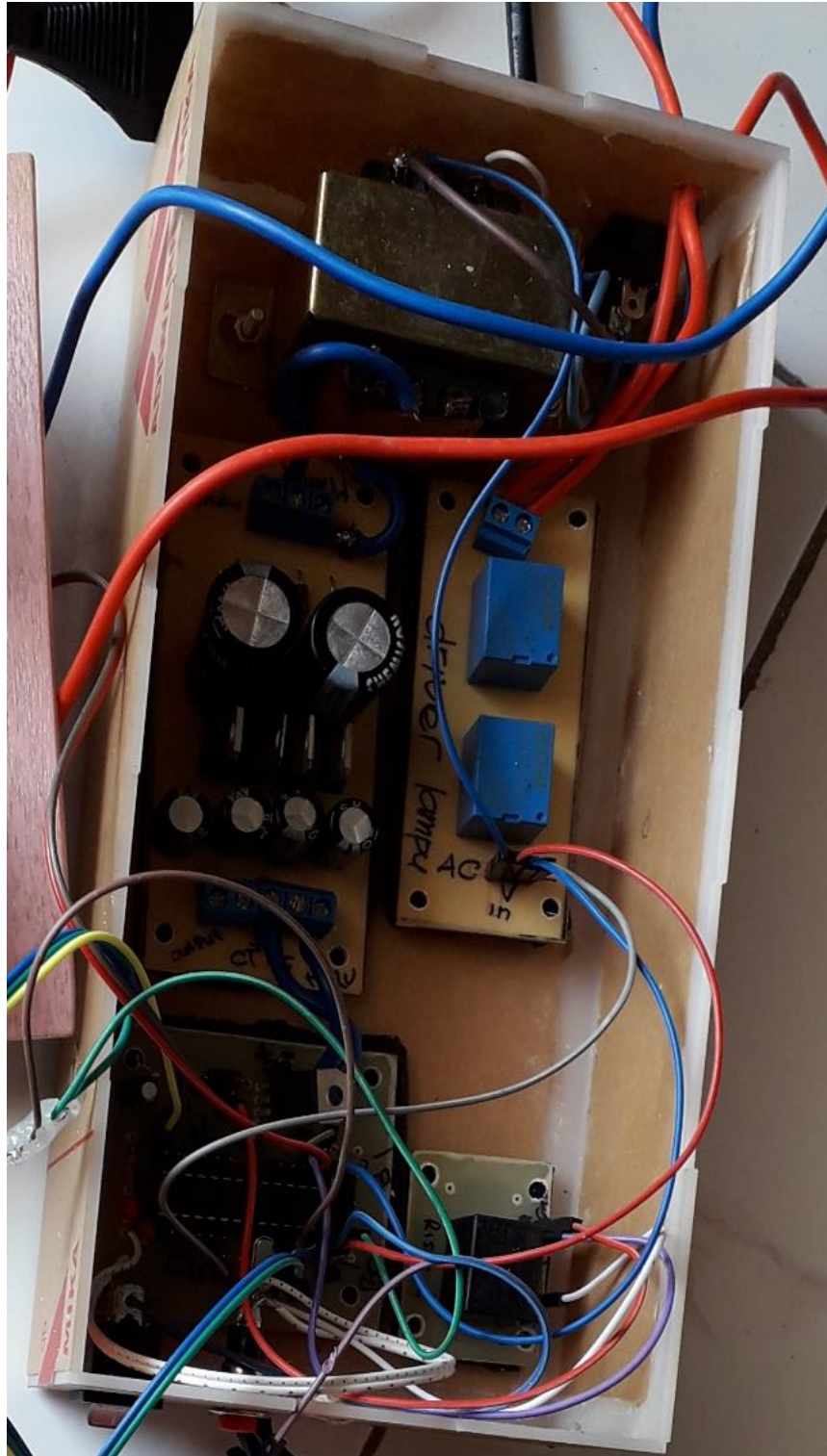


Keterangan :

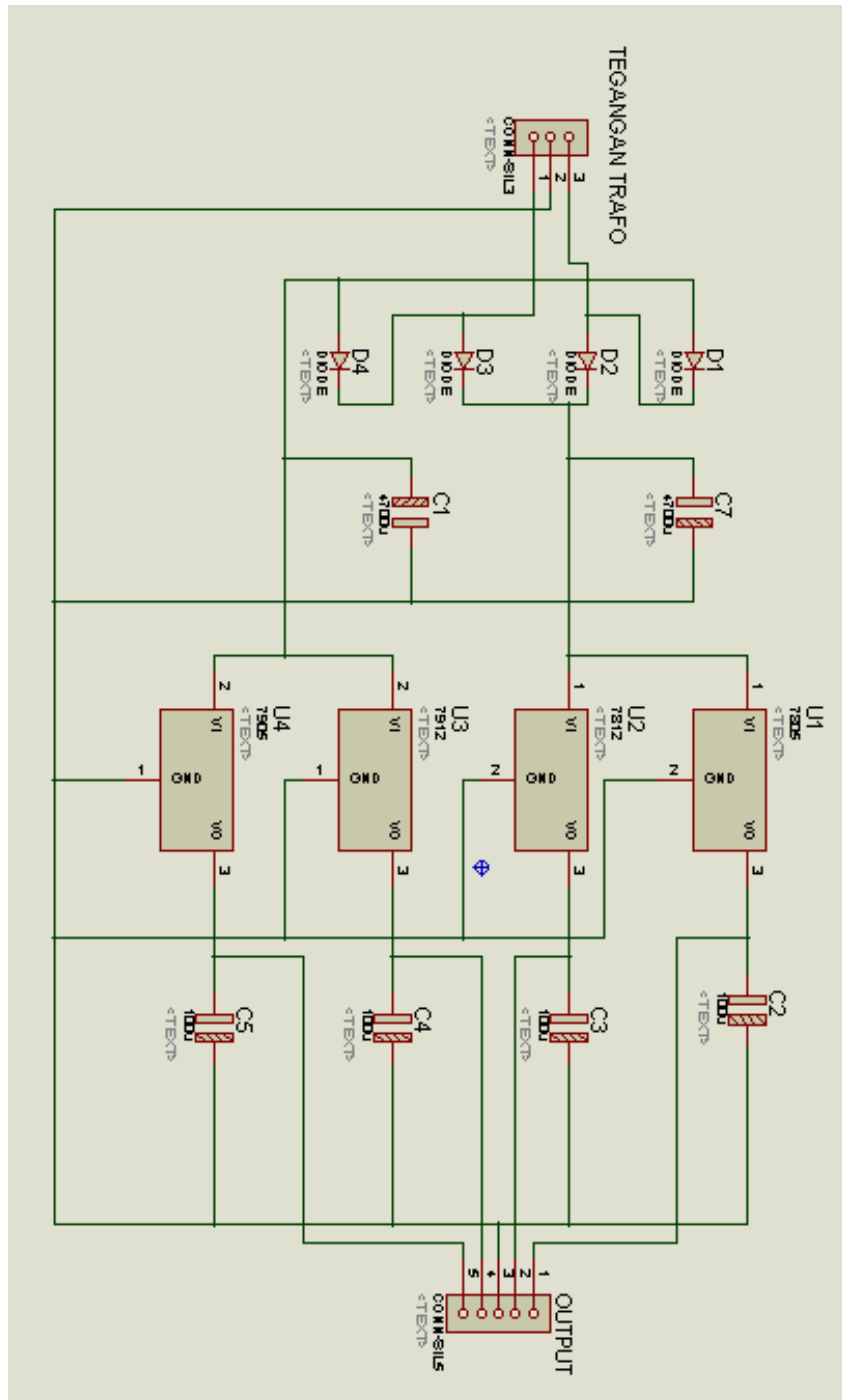
1 : Penempatan sensor magnet pada pintu.

2 : Penempatan sensor PIR pada pintu gerbang / utama.

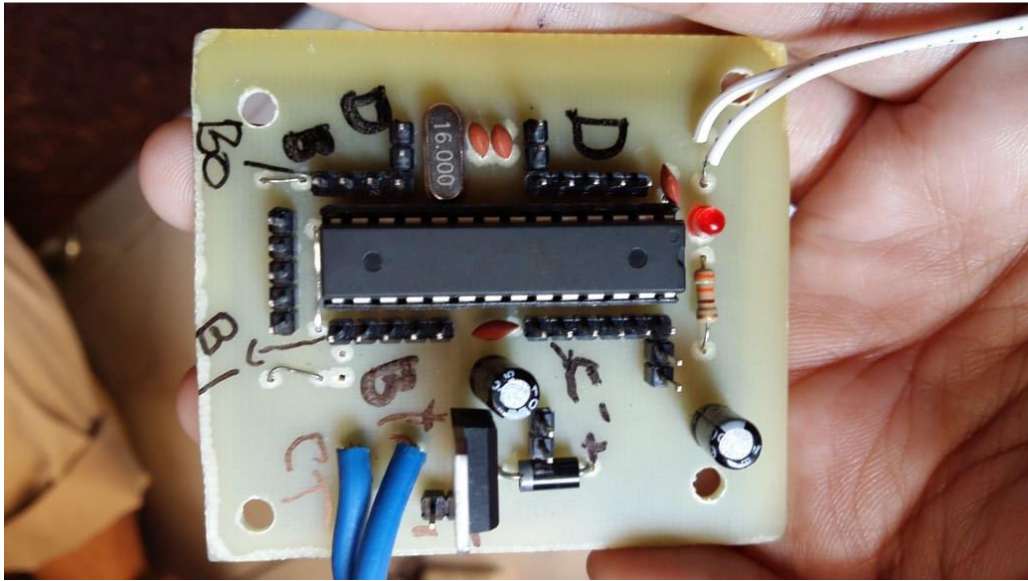
3. KOMPONEN DAN SKEMATIK ALAT



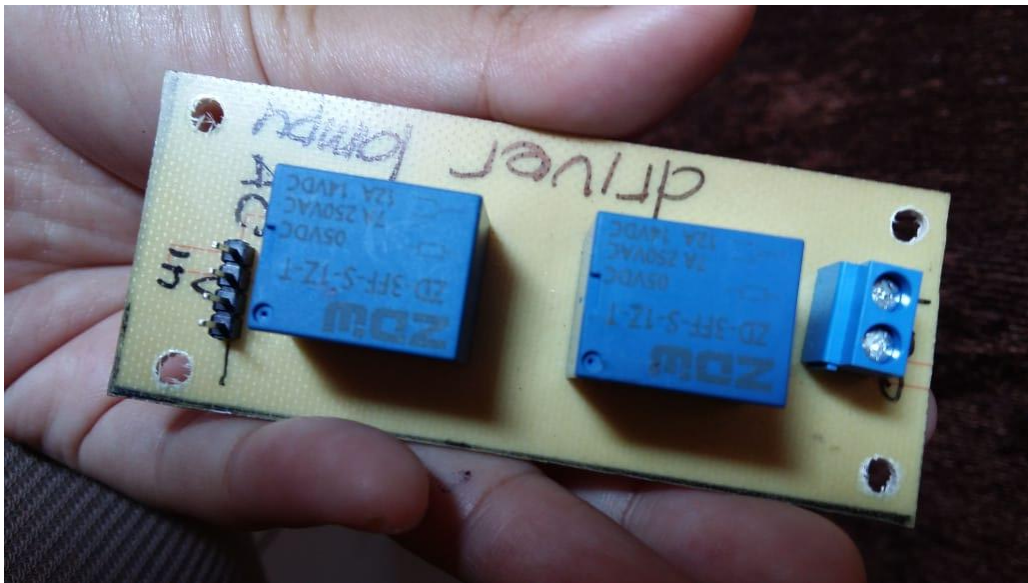
Gambar bagian alat keseluruhan



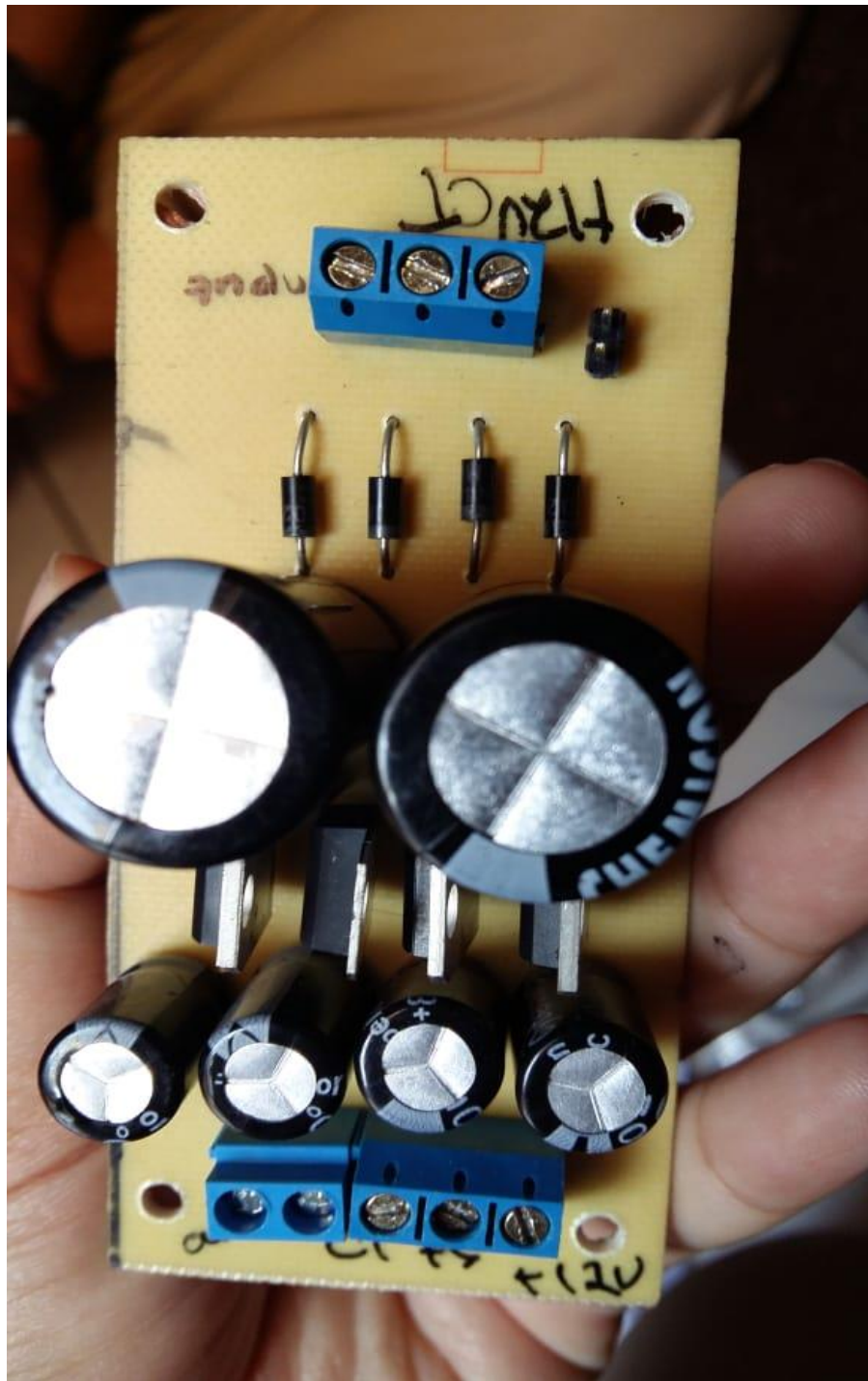
Gambar Skematik *Power Supply*



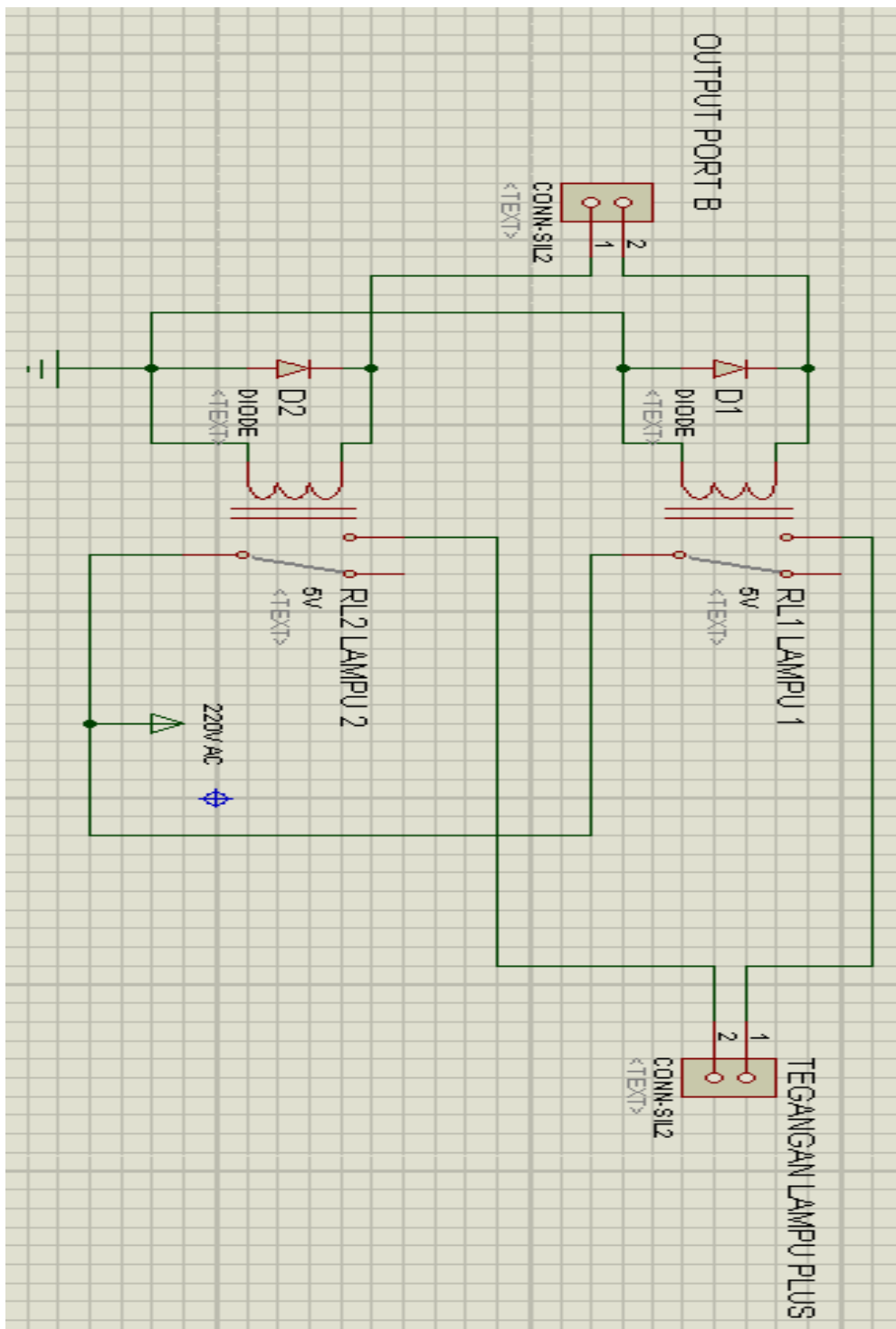
Gambar Minimum Sistem



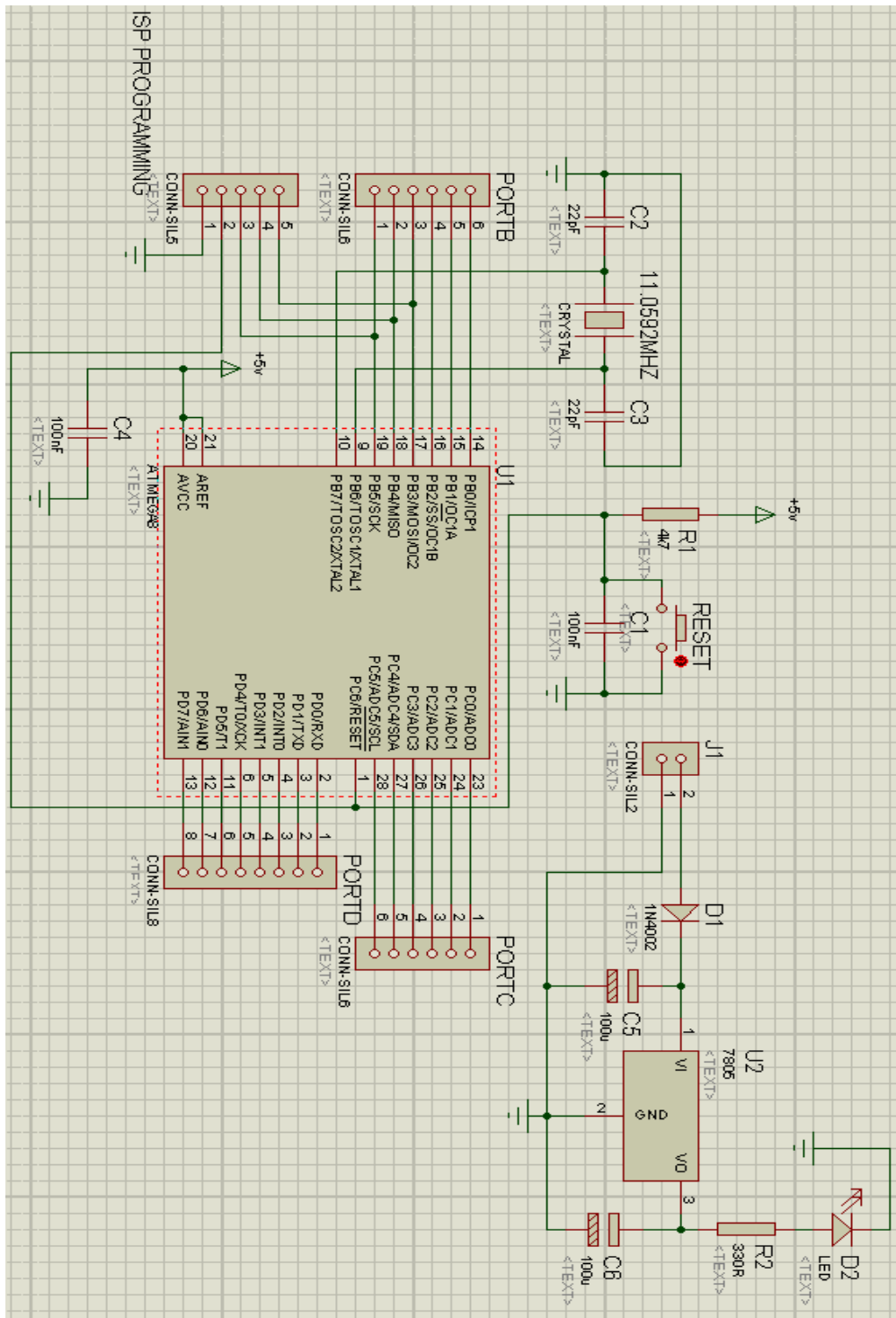
Gambar Driver Lampu



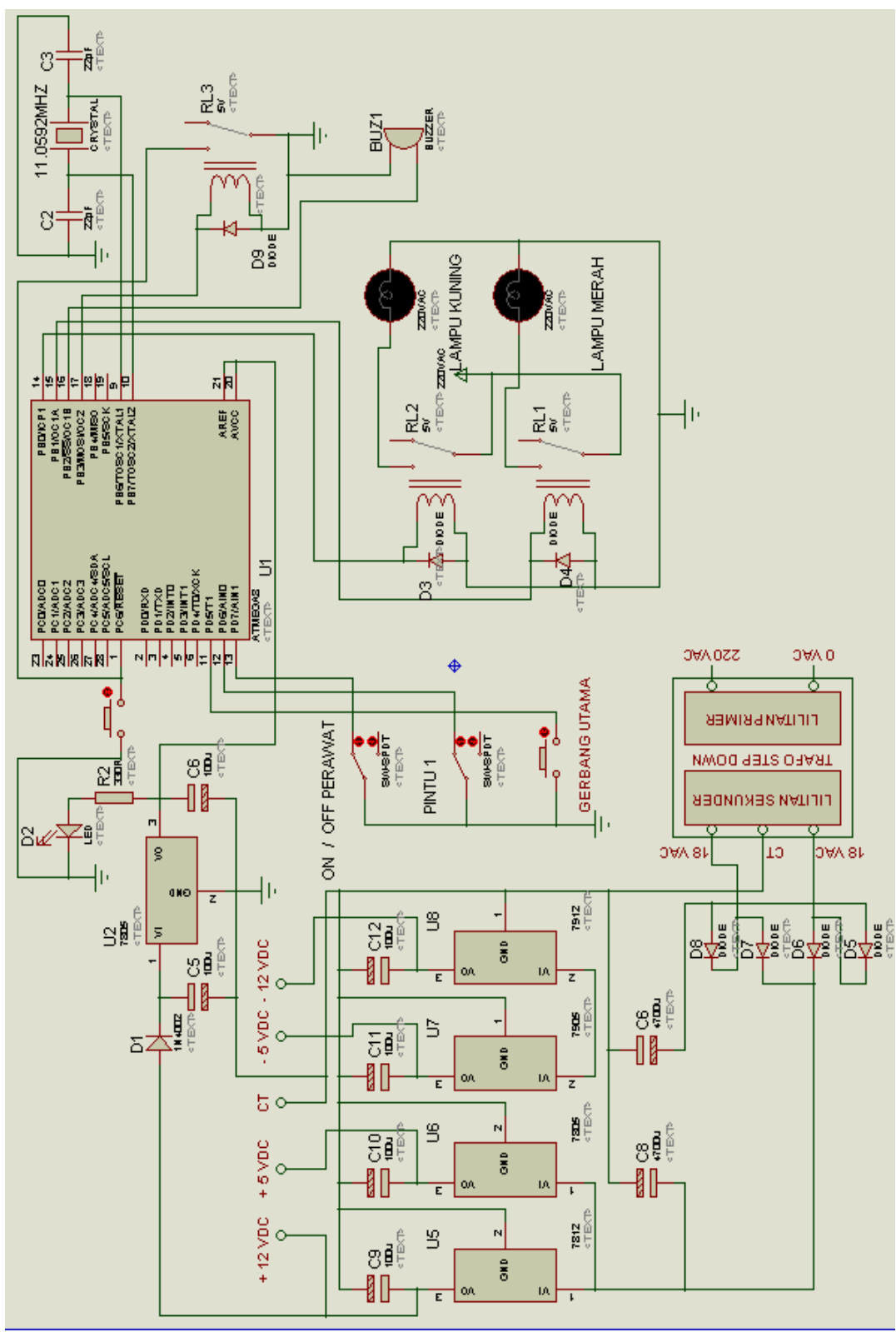
Gambar *Power Supply*



Gambar Skematik *Driver* Lampu



Gambar Skematik *Minimum Sistem*



Gambar rangkaian keseluruhan

Penjelasan Gambar rangkaian keseluruhan

Kabel PLN yang di sambungkan ke stop kontak membuat tegangan listrik mengalir menuju alat. Komponen yang pertama kali *aktif* adalah trafo *step down*. Trafo *step down* disini berfungsi sebagai penurun tegangan AC. Dapat dilihat pada gambar diatas, tegangan AC 220 VAC yang masuk pada lilitan primer akan diturunkan menjadi maksimal 18 VAC (tergantung dan besar ampere pada trafo). Tegangan AC bisa diturunkan dikarenakan lilitan sekunder pada trafo lebih sedikit daripada jumlah lilitan primer. Setelah melewati trafo, listrik akan di searahkan oleh dioda, besar ampere dioda menyesuaikan besar ampere pada trafo (dapat dilihat pada spesifikasi), dengan prinsip arus / tegangan masuk tidak boleh melewati di atas batas range yang telah ada di komponen, maka alat akan aman. Setelah disearahkan maka akan dilandaikan oleh kapasitor dan di stabilkan oleh regulator.

Di *power supplay* penulis, penulis menggunakan 7805 dan 7812, hal ini di lakukan apa bila ada regulator yang tidak berfungsi maka dengan mudah dapat diganti. Setelah melewati *power supplay* maka listrik akan menuju *microcontroller* akan tetapi harus melewati regulator 7805 untuk menstabilkan kembali listrik yang akan masuk ke *microcontroller* dan disebarkan ke blok-blok diagram alat.

Pada blok sensor PIR, data akan bertegangan 0 VDC apa bila sensor tidak mendeteksi adanya gerakan object. Sebaliknya sensor akan mendapatkan tegangan 3,3 VDC tanpa beban, apabila adanya beban (resistor, transistor, relay dll) maka tegangan di data menjadi 0,7 VDC.

Pada blok diagram sensor magnet prinsip kerjanya sama halnya saklar, sensor magnet akan menyambung apabila ada magnet yang mendekat dan adanya

persamaan kutub, maka akan terjadi tarik menarik antar besi yang ada di dalam sensor magnet. Sensor magnet dapat dialiri tegangan positif atau ground. Untuk saklar *ON / OFF* perawat dapat di aliri tegangan ground atau positif. Akan tetapi disini penulis menggunakan tegangan ground untuk mengalirinya.