

## **BAB III**

### **METODE DAN PERANCANGAN PENELITIAN**

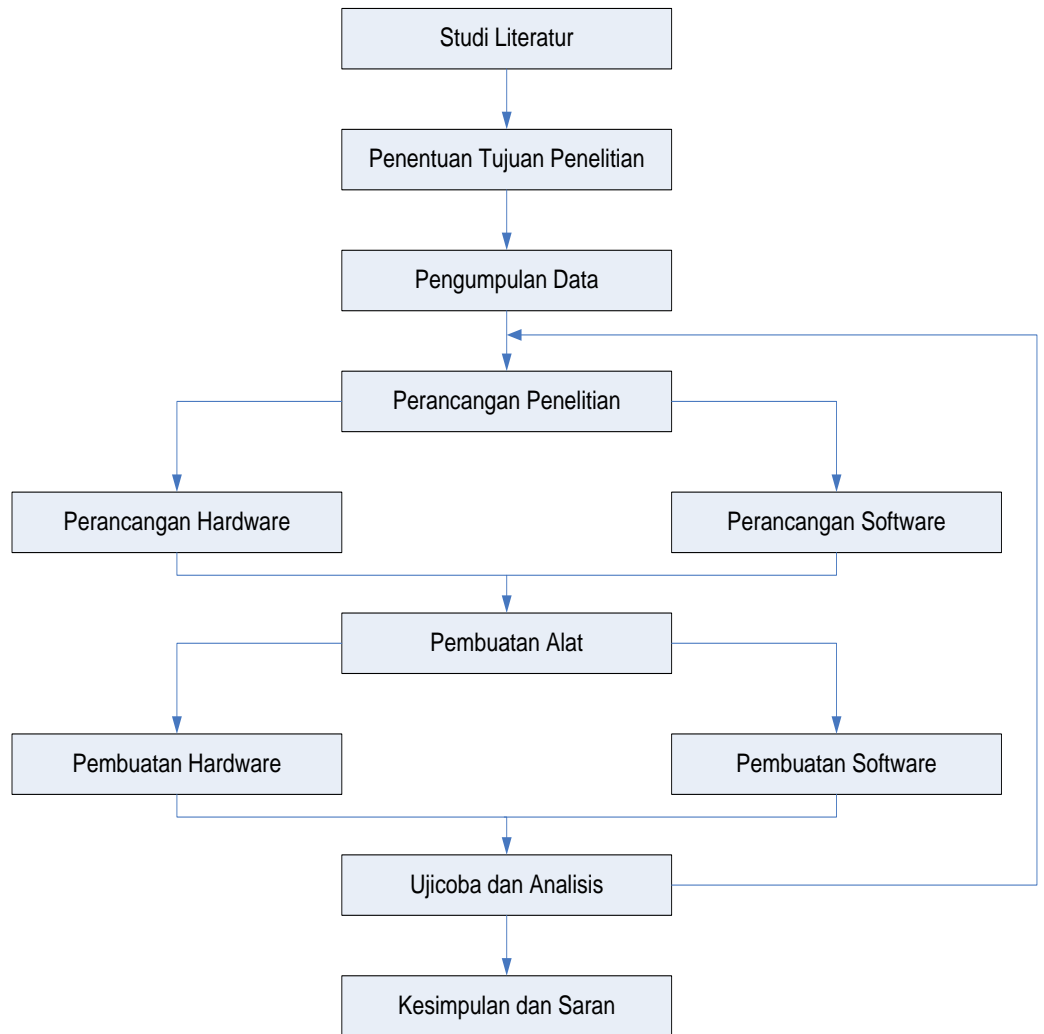
Dalam penelitian ini menggunakan metode rancang bangun, yang diawali dari pembuata prototype rumah kaca dengan ukuran panjang 1 m, lebar 60 cm dan tinggi 75 cm, selanjutnya dibuatlah sistem kendali intensitas cahaya sesuai dengan kebutuhan pada proses budidaya bunga krisan yang mengacu pada data litbang sulut, dimana kebutuhan intensitas cahaya tambahan di malam hari sebesar 100 lux. Setelah sistem selesai dibuat maka ditempatkan beberapa polibek bunga krisan dalam miniature rumah kaca tersebut, dan dilakukan pengambilan data intensitas cahaya dengan luxmeter. Untuk lebih jelasnya akan dibahas mendetail pada sub bab implementasi alat.

#### **3.1.Lokasi Dan Waktu**

- Lokasi : Laboratorium Perancangan dan M&R Elektronika Politeknik  
Negeri Manado, Lokasi budidaya bunga krisan di Tomohon.
- Waktu : Januari 2013 – Juni 2013

#### **3.2.Tahap-tahap Penelitian**

Adapun Tahapan penelitian yang akan dilakukan dapat dilihat pada bagan kotak gambar 3.1 berikut :



**Gambar 3.1. Bagan kotak tahapan penelitian**

Adapun tahapan tahapannya sebagai berikut :

1. *Studi Literatur*

Tahap ini dilakukan untuk mencari informasi sehubungan dengan kontrol intensitas cahaya sehubungan dengan proses pertumbuhan tanaman bunga krisan.

## 2. *Penentuan Tujuan Penulisan*

Tahap penentuan tujuan penelitian dilakukan untuk mengetahui arah pembuatan thesis ini.

## 3. *Pengumpulan Data*

Pada tahap ini dilakukan proses pengumpulan data, dimana data yang diambil adalah data besarnya arus listrik yang di konsumsi oleh lampu dengan menggunakan ampere meter, dan data intensitas cahaya yang diukur dengan menggunakan lux meter, serta data intensitas cahaya yang dibutuhkan oleh bunga krisan dalam proses pertumbuhannya.

## 4. *Perancangan penelitian*

Terdapat 2 bagian didalam tahap perancangan cepat yaitu :

### a. Perancangan Hardware

Perancangan hardware bertujuan untuk merancang peralatan/rangkaian pendukung untuk sistem yang akan dibuat.

### b. Perancangan Software

Perancangan software dilakukan untuk memudahkan didalam pembuatan software nanti.

## 5. *Tahap Pembuatan*

Terdapat 3 bagian didalam tahap pembuatan yaitu :

### a. Pembuatan Hardware

Pembuatan hardware merupakan proses untuk membuat rangkaian pendukung untuk sistem yang akan dibuat.

b. Pembuatan Software

Pembuatan software merupakan proses pembuatan program untuk sistem yang akan dibuat.

6. *Ujicoba dan analisis*

Tahap pengujian dilakukan untuk menguji kerja dari keseluruhan sistem, yang mencakup :

1. Pengujian sistem kontrol intensitas cahaya
2. Pengujian terhadap objek yang diuji didalam hal ini yaitu pertumbuhan bunga krisan terhadap pengaturan intensitas cahaya yang diberikan.

jika sistem yang diuji belum sesuai, maka kembali ke tahap pembuatan.

Tahap analisa dilakukan untuk menganalisa hasil pengujian dari sistem, apakah sistem yang dibuat tersebut telah sesuai dengan apa yang diharapkan.

Jika sistem yang dibuat belum sesuai, maka kembali ke tahap pengujian.

7. *Kesimpulan dan Saran*

Kesimpulan dan saran merupakan tahapan akhir dari keseluruhan sistem yang akan dibuat, dimana kesimpulan berisikan hal-hal yang dianggap pokok didalam proses pembuatan sistem, dan saran berisikan hal-hal yang

merupakan masukan dari pengguna sistem demi kesempurnaan sistem yang dibuat.

### **3.3. Alat dan bahan yang digunakan**

Alat dan bahan yang digunakan pada penelitian ini meliputi :

- ✓ Lux meter, yaitu alat untuk mengukur intensitas cahaya
- ✓ Lampu hemat energi 20 Watt
- ✓ 1 Set Mikrokontroler Arduino 328
- ✓ 1 set Mikrokontroler ATmega 8535
- ✓ LDR
- ✓ SCR
- ✓ Potensiometer
- ✓ Diac
- ✓ Kabel rangkaian
- ✓ Resistor
- ✓ Kapasitor
- ✓ Kayu dan plastic untuk pembuatan miniature rumah kaca

### **3.4. Rancangan Penelitian**

#### **3.4.1. Arduino Mikrokontroler AVR Tiper ATmega328**

Arduino merupakan kombinasi perangkat keras dan perangkat lunak yang bersifat *open source* .Board ini memiliki 14 digital input / output pin (dimana 6 pin

dapat digunakan sebagai output PWM), 6 input analog, 16 MHz osilator kristal, koneksi USB, jack listrik tombol reset. Pin-pin ini berisi semua yang diperlukan untuk mendukung mikrokontroler, hanya terhubung ke komputer dengan kabel USB atau sumber tegangan bisa didapat dari adaptor AC-DC atau baterai untuk menggunakannya.



**Gambar 3.2 Board Arduino Uno**

[Sumber : <http://arduino.cc/en/Main/arduinoBoardUno>]

**Tabel 3.1 Deskripsi Arduino Uno**

Mikrokontroller	Atmega328
Operasi Voltage	5V
Input Voltage	7-12 V (Rekomendasi)
Input Voltage	6-20 V (limits)
I/O	14 pin (6 pin untuk PWM)
Arus	50 mA
Flash Memory	32KB
Bootloader	SRAM 2 KB
EEPROM	1 KB
Kecepatan	16 Mhz

Uno Arduino dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Sumber listrik dipilih secara otomatis. Eksternal (nonUSB) daya dapat datang baik dari AC-DC adaptor atau baterai. Adaptor ini dapat dihubungkan dengan cara menghubungkannya plug pusat-positif 2.1mm ke dalam board colokan listrik. Lead dari baterai dapat dimasukkan ke dalam header pin Gnd dan Vin dari konektor Power.

Board dapat beroperasi pada pasokan daya dari 6 - 20 volt. Jika diberikan dengan kurang dari 7V, bagaimanapun, pin 5V dapat menyuplai kurang dari 5 volt dan board mungkin tidak stabil. Jika menggunakan lebih dari 12V, regulator tegangan bisa panas dan merusak board. Rentang yang dianjurkan adalah 7 - 12 volt.

Pin catu daya adalah sebagai berikut:

- ✍ VIN. Tegangan input ke board Arduino ketika menggunakan sumber daya eksternal (sebagai lawan dari 5 volt dari koneksi USB atau sumber daya lainnya diatur). Anda dapat menyediakan tegangan melalui pin ini, atau, jika memasok tegangan melalui colokan listrik, mengaksesnya melalui pin ini.
- ✍ 5V. Catu daya diatur digunakan untuk daya mikrokontroler dan komponen lainnya di board. Hal ini dapat terjadi baik dari VIN melalui regulator on-board, atau diberikan oleh USB .
- ✍ 3,3 volt pasokan yang dihasilkan oleh regulator on-board. Menarik arus maksimum adalah 50 mA.
- ✍ GND

ATmega328 ini memiliki 32 KB dengan 0,5 KB digunakan untuk loading file. Ia juga memiliki 2 KB dari SRAM dan 1 KB dari EEPROM. Masing-masing dari 14 pin digital pada Uno dapat digunakan sebagai input atau output, menggunakan fungsi `pinMode()`, `digitalWrite()`, dan `digitalRead()`. Mereka beroperasi di 5 volt. Setiap pin dapat memberikan atau menerima maksimum 40 mA dan memiliki resistor pull-up internal dari 20-50 KΩ. Selain itu, beberapa pin memiliki fungsi khusus:

- ✓ Serial: 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) data TTL serial. Pin ini terhubung ke pin yang sesuai dari chip ATmega8U2 USB-to-Serial TTL.
- ✓ Eksternal Interupsi: 2 dan 3. Pin ini dapat dikonfigurasi untuk memicu interupsi pada nilai yang rendah, tepi naik atau jatuh, atau perubahan nilai. Lihat `attachInterrupt()` fungsi untuk rincian.
- ✓ PWM: 3, 5, 6, 9, 10, dan 11. Menyediakan 8-bit output PWM dengan `analogWrite()` fungsi.
- ✓ SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Pin ini mendukung komunikasi SPI menggunakan perpustakaan SPI.
- ✓ LED: 13. Ada built-in LED terhubung ke pin digital 13. Ketika pin adalah nilai TINGGI, LED menyala, ketika pin adalah RENDAH, itu off.

Uno memiliki 6 input analog, diberi label A0 melalui A5, masing-masing menyediakan 10 bit resolusi yaitu 1024 nilai yang berbeda. Secara default sistem mengukur dari tanah sampai 5 volt.



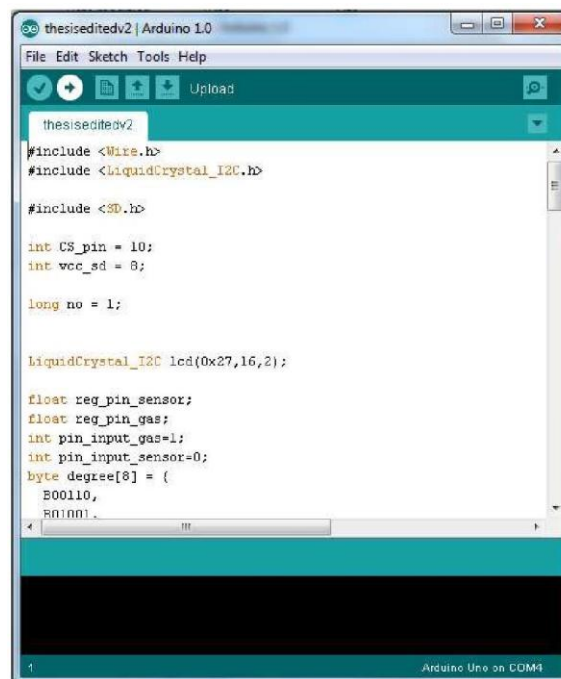
- TWI: A4 atau SDA pin dan A5 atau SCL pin. Mendukung komunikasi TWI
- Aref. Referensi tegangan untuk input analog. Digunakan dengan `analogReference()`.
- Reset.

Uno Arduino memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain, atau mikrokontroler lain. ATmega328 ini menyediakan UART TTL (5V) komunikasi serial, yang tersedia pada pin digital 0 (RX) dan 1 (TX). Sebuah ATmega16U2 pada saluran board ini komunikasi serial melalui USB dan muncul sebagai com port virtual untuk perangkat lunak pada komputer. Firmware Arduino menggunakan USB driver standar COM, dan tidak ada driver eksternal yang dibutuhkan. Namun, pada Windows, file. Inf diperlukan. Perangkat lunak Arduino termasuk monitor serial yang memungkinkan data sederhana yang akan dikirim ke board Arduino. RX dan TX LED di board akan berkedip ketika data sedang dikirim melalui chip USB-to-serial dan koneksi USB ke komputer. ATmega328 ini juga mendukung komunikasi I2C (TWI) dan SPI. Fungsi ini digunakan untuk melakukan komunikasi interface pada sistem.

Uno Arduino dapat diprogram dengan perangkat lunak Arduino. Pilih Arduino Uno dari Tool lalu sesuaikan dengan mikrokontroler yang digunakan. Pada ATmega328 pada Uno Arduino memiliki bootloader yang memungkinkan Anda untuk meng-upload program baru untuk itu tanpa menggunakan programmer hardware eksternal. Ini berkomunikasi menggunakan protokol dari bahasa C.

Sistem dapat menggunakan perangkat lunak FLIP Atmel (Windows) atau programmer DFU (Mac OS X dan Linux) untuk memuat firmware baru. Atau Anda dapat menggunakan header ISP dengan programmer eksternal .

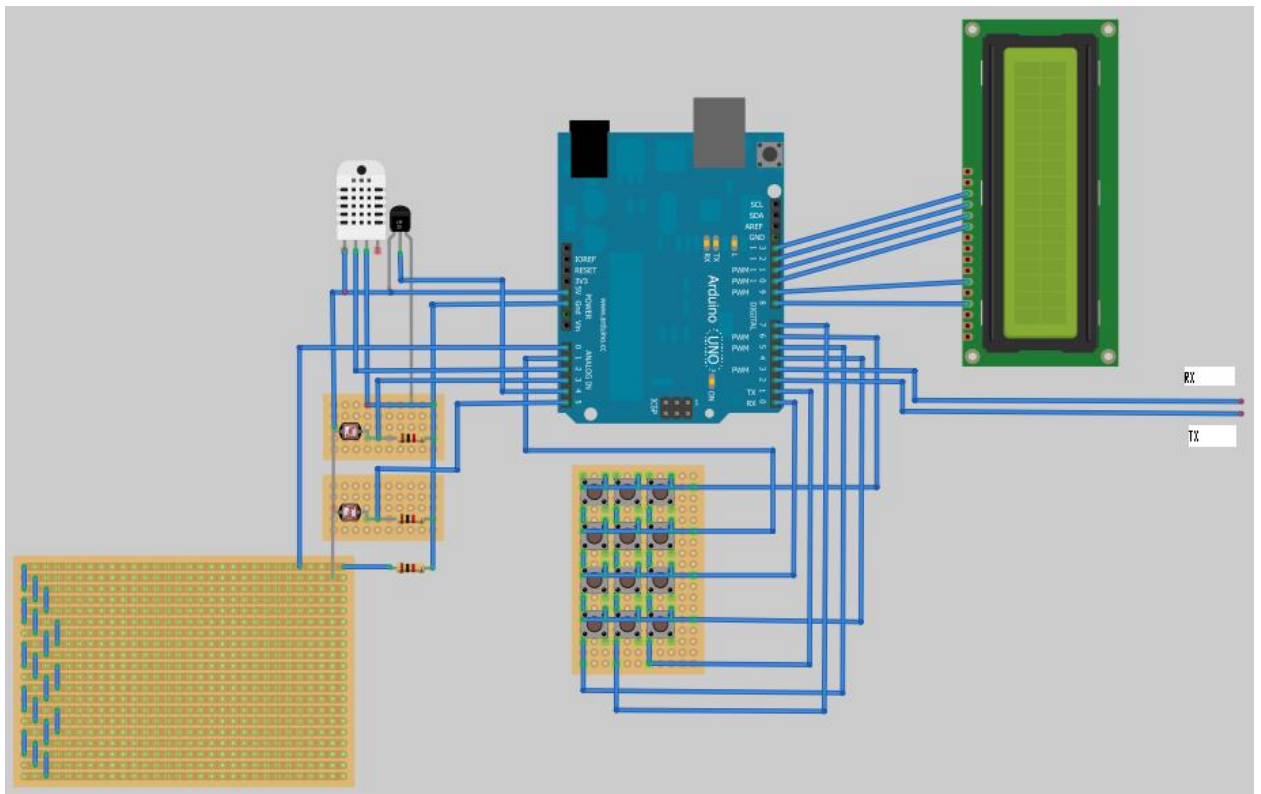
Lingkungan open-source Arduino memudahkan untuk menulis kode dan meng-upload ke board Arduino. Ini berjalan pada Windows, Mac OS X, dan Linux. Berdasarkan Pengolahan, avr-gcc, dan perangkat lunak sumber terbuka lainnya. [Sujadi, MT, 2005, *Teori dan Aplikasi Mikrokontroler*]



**Gambar 3.3 Tampilan Framework Arduino UNO**

[Sumber : Sujadi, MT, 2005, *Teori dan Aplikasi Mikrokontroler*]

Pada sistem yang dibangun, mikrokontroler AVR 328 Arduino difungsikan untuk LCD, Keypad, Sensor Suhu, Sensor Kelembapan, LDR, dan sensor curah hujan.



**Gbr 3.4. Rangkaian Mikrokontroler AVR 328 ARDUINO pada rumah kaca**

Berikut ini adalah program yang dibuat sehubungan dengan kendali LCD, Keypad, Sensor Suhu, Sensor Kelembapan, LDR, dan sensor curah hujan.

```
#include <LiquidCrystal.h>
#include <Keypad.h>
#include <EEPROM.h>
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2, 3);
// software serial #2: TX = digital pin 4, RX = digital pin 5
```

```

LiquidCrystal lcd(8, 9, 10, 11, 12, 13);
int sensor_cahaya,sensor_suhu,sensor_ph,sensor_hujan,sensor_hujan_ldr;
int key,max_cahaya,max_suhu,max_ph,data_cahaya,data_suhu,data_ph;
int ratusan,puluhan,satuan=0;
const byte ROWS = 4; //four rows
const byte COLS = 3; //four columns
char keys[ROWS][COLS] = {{1,2,3},{4,5,6},{7,8,9},{10,12,11}};
char fasa;
byte rowPins[ROWS] = {6,A1,0,4}; //connect to the row pinouts of the keypad
byte colPins[COLS] = {5,7,1}; //connect to the column pinouts of the keypad
int n,buka,tutup,lama,panas,lampu;
int terang=20;
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );

void setup(){
  lcd.begin(16, 2);
  BACA_MEMORIY();
  mySerial.begin(57600);
}
void READ_ALL_SENSOR() {lcd.begin(16, 2);
  lcd.clear();
  lcd.setCursor(0,0);lcd.print("FLUX: SUHU: PH:");
  sensor_cahaya = (analogRead(A3)/3)+10;
  sensor_suhu = (analogRead(A4)/2)-70;
  sensor_hujan_ldr = analogRead(A5)/2;
  sensor_hujan = analogRead(A0);
  sensor_ph = analogRead(A2)/8;
  lcd.setCursor(0,1);lcd.print(sensor_cahaya,DEC);
  lcd.setCursor(6,1);lcd.print(sensor_suhu,DEC);
  lcd.setCursor(12,1);lcd.print(sensor_ph,DEC);}
void BACA_MEMORIY(){buka = EEPROM.read(0);
  tutup = EEPROM.read(1);}
void TULIS_MEMORIY(){EEPROM.write(0, buka);
  EEPROM.write(1, tutup);}
void ATAP_BUKA() {if(buka==0){lcd.clear();
  lcd.setCursor(0,0);lcd.print("BUKA ATAP.....");
  for(byte
x=0;x<31;x++){mySerial.write(27);delay(58);mySerial.write(36);delay(30);buka=1;t
utup=0;TULIS_MEMORIY();}lcd.begin(16, 2);}}
void SIRAM()
{if(lama==0){mySerial.write(28);delay(60);mySerial.write(26);delay(40);mySerial.w
rite(36);lama=50;lcd.begin(16, 2);}lama--;}

```

```

void LAMPU_TERANG(){if(lampu==0){LAMPU_ON();lampu=1;}else

{ if(terang<40){ mySerial.write(29);delay(40);mySerial.write(36);delay(30);terang++;l
cd.begin(16, 2);}}}
void LAMPU_REDUP()
{ if(terang>0){ mySerial.write(30);delay(40);mySerial.write(36);delay(30);terang--
;lcd.begin(16, 2);lampu=1;}
    else{ if((lampu==1)&&(terang==0)){ LAMPU_OFF();lampu=0;}}
    }
void LAMPU_ON()  { mySerial.write(31);delay(100);mySerial.write(36);for(byte
x=0;x<15;x++){ READ_ALL_SENSOR();delay(100);}lcd.begin(16, 2);}
void LAMPU_OFF() { mySerial.write(32);delay(100);mySerial.write(36);for(byte
x=0;x<15;x++){ READ_ALL_SENSOR();delay(100);}lcd.begin(16, 2);}
void Relay8()
{ mySerial.write(33);delay(100);mySerial.write(36);delay(100);lcd.begin(16, 2);}
void PANAS()     { mySerial.write(34);READ_ALL_SENSOR();
    delay(300); lcd.begin(16, 2);panas=1;}
void ATAP_TUTUP() { if(tutup==0){ lcd.clear();
    lcd.setCursor(0,0);lcd.print("TUTUP ATAP.....");
    for(byte
x=0;x<34;x++){ mySerial.write(35);delay(45);mySerial.write(36);delay(50);tutup=1;b
uka=0;TULIS_MEMORIY();}lcd.begin(16, 2);}}
void tulis_isi_set_poit(){
    lcd.clear();lcd.setCursor(0,0);lcd.print("ATUR SET POINT");delay(1500);}
void tulis_max() {
    lcd.clear();lcd.setCursor(0,0);lcd.print("MAKSIMAL");}
void tulis_flux() {
    lcd.clear();lcd.setCursor(0,0);lcd.print("SET FLUX :");lcd.setCursor(0,1);lcd.print("
Candela");}
void tulis_suhu() {
    lcd.clear();lcd.setCursor(0,0);lcd.print("SET SUHU :");lcd.setCursor(0,1);lcd.print("
'C");}
void tulis_ph() {
    lcd.clear();lcd.setCursor(0,0);lcd.print("SET PH :");lcd.setCursor(0,1);lcd.print("
%");}
void tulis_flux_error(){

tulis_max();lcd.setCursor(0,1);lcd.print(max_cahaya);lcd.setCursor(5,1);lcd.print("Ca
ndela");}
void tulis_suhu_error(){

tulis_max();lcd.setCursor(0,1);lcd.print(max_suhu);lcd.setCursor(5,1);lcd.print("'C");
}

```

```

void tulis_ph_eror(){
    tulis_max();lcd.setCursor(0,1);lcd.print(max_ph);lcd.setCursor(5,1);lcd.print("%");}

void loop() {
    start:
    tulis_isi_set_poit();
    max_cahaya=250;
    max_suhu=100;
    max_ph=120;
    ulang:
    tulis_flux();
    lcd.setCursor(2,1);lcd.print(data_cahaya);delay(200);
    lagi:
    while (fasa==0) {
        key = keypad.getKey();
        if (key != NO_KEY) {n++;
            if(key==12) {if(n==1){n=0;}key=0;}
            if(key==10) {n=n-1;goto lagi;}
            if(key==11) {if (data_cahaya==0) {tulis_isi_set_poit();n=0;goto
ulang;} else
                if (data_cahaya>(max_cahaya))
{tulis_flux_eror();delay(1500);n=0;data_cahaya=0;goto ulang;}
                else {n=0;fasa=1;goto ulang1;}}
            if(n==1){tulis_flux();satuan=key;data_cahaya=satuan;} else

if(n==2){puluhan=satuan*10;satuan=key;data_cahaya=puluhan+satuan;} else

if(n==3){ratusan=puluhan*10;puluhan=satuan*10;satuan=key;data_cahaya=ratusan+
puluhan+satuan;}
            lcd.setCursor(1,1);lcd.print(data_cahaya);delay(200); }}
    ulang1:
    tulis_suhu();
    lcd.setCursor(2,1);lcd.print(data_suhu);delay(200);
    while (fasa==1) {
        key = keypad.getKey();
        if (key != NO_KEY) {n++;
            if(key==12) {if(n==1){n=0;}key=0;}
            if(key==10) {n=0;fasa=0;goto ulang;}
            if(key==11) {if (data_suhu==0) {tulis_isi_set_poit();n=0;goto
ulang1;} else
                if (data_suhu>(max_suhu))
{tulis_suhu_eror();delay(1500);n=0;data_suhu=0;goto ulang1;}

```

```

        else {n=0;fasa=2;goto ulang2;}}
        if(n==1){tulis_suhu();satuan=key;data_suhu=satuan;} else

if(n==2){puluhan=satuan*10;satuan=key;data_suhu=puluhan+satuan;} else

if(n==3){ratusan=puluhan*10;puluhan=satuan*10;satuan=key;data_suhu=ratusan+pu
luhan+satuan;}

        lcd.setCursor(1,1);lcd.print(data_suhu);delay(200); }}

        ulang2:
        tulis_ph();
        lcd.setCursor(2,1);lcd.print(data_ph);delay(200);
        while (fasa==2) {
        key = keypad.getKey();
        if (key != NO_KEY)      {n++;
                                if(key==12) {if(n==1){n=0;}key=0;}
                                if(key==10) {n=0;fasa=1;goto ulang1;}
                                if(key==11) {if (data_ph==0) {tulis_isi_set_poit();n=0;goto
ulang2;} else
                                                if (data_ph>(max_ph))
{tulis_ph_error();delay(1500);n=0;data_ph=0;goto ulang2;}
                                                else {n=0;fasa=3;goto proses;}}
                                if(n==1){tulis_ph();satuan=key;data_ph=satuan;} else

if(n==2){puluhan=satuan*10;satuan=key;data_ph=puluhan+satuan;} else

if(n==3){ratusan=puluhan*10;puluhan=satuan*10;satuan=key;data_ph=ratusan+pu
luhan+satuan;}

                                lcd.setCursor(1,1);lcd.print(data_ph);delay(200); }}

        proses:
        lcd.clear();
        lcd.setCursor(0,0);lcd.print("SET POINT FLUX:");
        lcd.setCursor(0,1);lcd.print((data_cahaya-4),DEC);lcd.print(" -
");lcd.print((data_cahaya+4),DEC);
        delay(2000);
        lcd.clear();
        lcd.setCursor(0,0);lcd.print("SET POINT SUHU:");
        lcd.setCursor(0,1);lcd.print(data_suhu,DEC);
        lcd.setCursor(5,1);lcd.print("C");
        delay(2000);
        lcd.clear();
        lcd.setCursor(0,0);lcd.print("SET POINT PH:");

```

```

    lcd.setCursor(0,1);lcd.print((data_ph-2),DEC);lcd.print(" %");lcd.print(" -
");lcd.print((data_ph+2),DEC);lcd.print(" %");
    delay(2000);
    lcd.clear();
    lcd.setCursor(0,0);lcd.print("RUNNING PROJECT...");
    lcd.setCursor(0,1);lcd.print("TEKAN '#' RESTART");
    delay(2000);
    while (fasa==3) {lcd.clear();
        lcd.setCursor(0,0);lcd.print("FLUX: SUHU: PH:");
        sensor_cahaya = (analogRead(A3)/3)-20;
        sensor_suhu = (analogRead(A4)/2)-70;
        sensor_hujan_ldr = analogRead(A5)/2;
        sensor_hujan = analogRead(A0);
        sensor_ph = analogRead(A2)/8;
        lcd.setCursor(0,1);lcd.print(sensor_cahaya,DEC);
        lcd.setCursor(6,1);lcd.print(sensor_suhu,DEC);
        lcd.setCursor(12,1);lcd.print(sensor_ph,DEC);
        if(sensor_cahaya>(data_cahaya+4)){LAMPU_REDUP();}
        if(sensor_cahaya<(data_cahaya-4)){LAMPU_TERANG();}

if(sensor_hujan>10){if(sensor_hujan_ldr<300){ATAP_BUKA();}else{ATAP_TUT
UP();}}
        if(sensor_hujan<10){ATAP_TUTUP();}
        if(sensor_suhu>data_suhu) {SIRAM();}
        if(sensor_suhu<data_suhu) {lama=0;}
        if(sensor_ph<(data_ph+2)) {PANAS();}
        if(sensor_ph>(data_ph-2)) {if(panas==1){mySerial.write(36);for(byte
x=0;x<20;x++){READ_ALL_SENSOR();delay(200);}panas=0;lcd.begin(16, 2);}}
        key = keypad.getKey();
        if (key != NO_KEY)
            {if(key==11) {fasa=0;goto start;}
            }
        delay(130);}

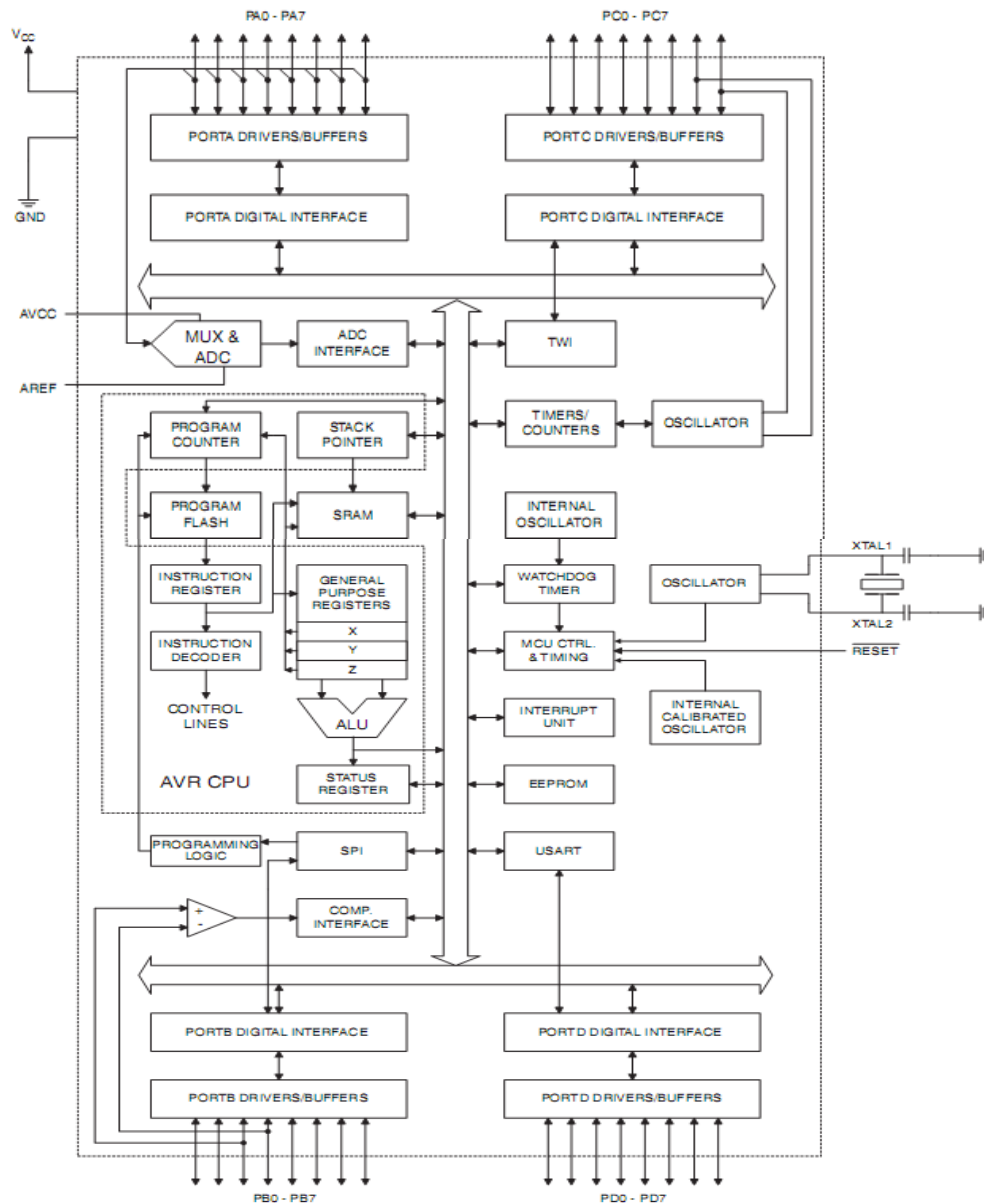
}

```



### 3.4.2. Mikrokontroler ATmega 8535

Mikrokontroler ATmega8535 merupakan mikrokontroler generasi AVR (*Alf and Vegard's Risk processor*).



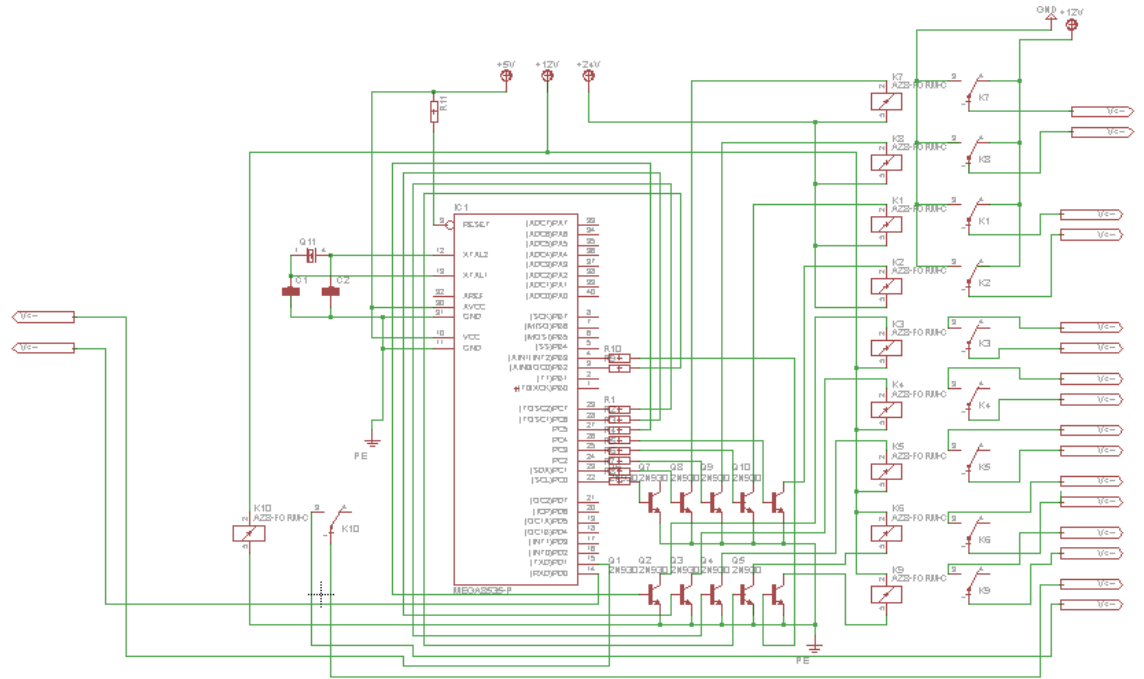
**Gambar 3.5 Diagram Blok Fungsional ATmega8535**  
(diambil dari data sheet ATmega8535)

Mikrokontroller AVR memiliki arsitektur RISC (*Reduced Instruction Set Computing*) 8 bit, dimana semua instruksi dikemas dalam kode 16-bit (*16-bits word*) dan sebagian besar instruksi dieksekusi dalam 1 siklus clock.

Gambar 3.5 memperlihatkan bahwa ATmega8535 memiliki bagian sebagai berikut :

1. Saluran I/O sebanyak 32 buah, yaitu Port A, Port B, Port C, dan Port D.
2. ADC 10 bit sebanyak 8 saluran.
3. Tiga buah *Timer/Counter* dengan kemampuan pembandingan.
4. CPU yang terdiri atas 32 buah register.
5. *Watchdog Timer* dengan osilator internal.
6. SRAM sebesar 512 byte.
7. Memori *Flash* sebesar 8 Kb dengan kemampuan *Read While Write*.
8. Unit interupsi internal dan eksternal.
9. Port antarmuka SPI.
10. EEPROM (*Electrically Erasable Programmable Read Only Memory*) sebesar 512 byte yang dapat diprogram saat operasi.
11. Antarmuka komparator analog.
12. Port USART untuk komunikasi serial dengan kecepatan maksimal 2,5 Mbps.
13. Sistem mikroprosesor 8 bit berbasis RISC dengan kecepatan maksimal 16 MHz.

Berikut ini adalah gambar rangkaian mikrokontroler AVR 8535 yang di aplikasikan pada sistem rumah kaca yang dibuat.



**Gambar 3.6. Desain Mikrokontroler AVR 8535 pada sistem**

Mikrokontroler tipe AVR8535 difungsikan untuk menggerakkan Driver Relay.

Berikut ini adalah program untuk menggerakkan driver relay :

```
#include<mega8535.h>
#include<stdio.h>
#include<delay.h>

=====

#define P0 PORTB.2
#define P1 PORTB.3 //motor
#define P2 PORTC.7
#define P3 PORTC.6
#define P4 PORTC.5
#define P5 PORTC.4
#define P6 PORTC.3
#define P7 PORTC.2
```

```

#define P8 PORTC.1
#define P9 PORTC.0

const long int osilator=1200000;
unsigned long int UBRR;
char data;
/*fungsi Inisialisasi UART*/
void InisialisasiUART ( unsigned long int baud_rate)
{
    UBRR=(osilator/(16*baud_rate))-1;
    UBRRH=UBRR;
    UBSR=UBRR>>8;
    UCSRB=0b00011000;
    UCSRC=0x8e;
}
void inisialisasi_io()
{
    DDRC=0xff;DDRB=0xff;
    PORTB=PORTC=0xff;PORTD=0xff;
}
void main(void){
    inisialisasi_io();
    InisialisasiUART(57600);
    P0=0;P1=0;P2=0;P3=0;P4=0;P5=0;P6=0;P7=0;P8=0;P9=0;
    while(1){
        data=getchar();
        if (data==58) {P0=1;P1=0;P2=0;P3=0;P4=0;P5=0;P6=0;P7=0;P8=0;P9=0;}
    else
        if (data==59) {P0=0;P1=1;P2=0;P3=0;P4=0;P5=0;P6=0;P7=0;P8=0;P9=0;}
    else
        if (data==60) {P0=0;P1=0;P2=1;P3=0;P4=0;P5=0;P6=0;P7=0;P8=0;P9=0;}
    else
        if (data==61) {P0=0;P1=0;P2=0;P3=1;P4=0;P5=0;P6=0;P7=0;P8=0;P9=0;}
    else
        if (data==62) {P0=0;P1=0;P2=0;P3=0;P4=1;P5=0;P6=0;P7=0;P8=0;P9=0;}
    else
        if (data==63) {P0=0;P1=0;P2=0;P3=0;P4=0;P5=1;P6=0;P7=0;P8=0;P9=0;}
    else
        if (data==64) {P0=0;P1=0;P2=0;P3=0;P4=0;P5=0;P6=1;P7=0;P8=0;P9=0;}
    else
        if (data==65) {P0=0;P1=0;P2=0;P3=0;P4=0;P5=0;P6=0;P7=1;P8=0;P9=0;}
    else

```

```

        if (data==66) {P0=0;P1=0;P2=0;P3=0;P4=0;P5=0;P6=0;P7=0;P8=1;P9=0;}
else
        if (data==67) {P0=0;P1=0;P2=0;P3=0;P4=0;P5=0;P6=0;P7=0;P8=0;P9=1;}
else
        {P0=0;P1=0;P2=0;P3=0;P4=0;P5=0;P6=0;P7=0;P8=0;P9=0;}

}
}

```

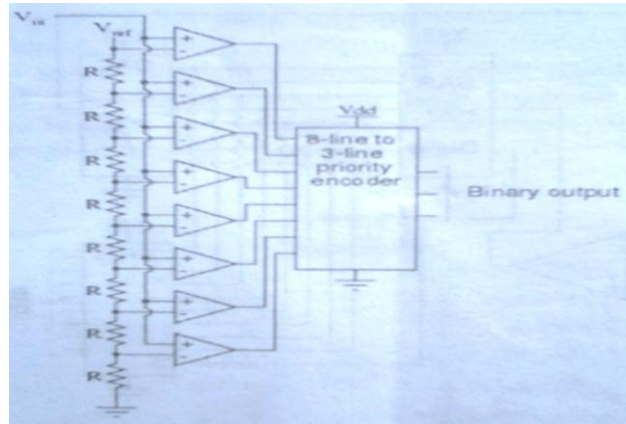
### 3.4.3. Pengkonversi Analog ke Digital (*Analog to Digital Converter*)

Rangkaian atau Chip ADC berfungsi untuk mengubah sinyal analog menjadi sinyal digital. Umumnya kita menggunakan Chip ADC 8 bit untuk mengubah rentang sinyal analog 0-5 V menjadi level digital 0-255 untuk adc 8 bit, meskipun saat ini sudah banyak adc yang mampu memproses data 12 bit.

Metode Konversi ADC terdapat 2 cara ;

#### 1. Flash ADC

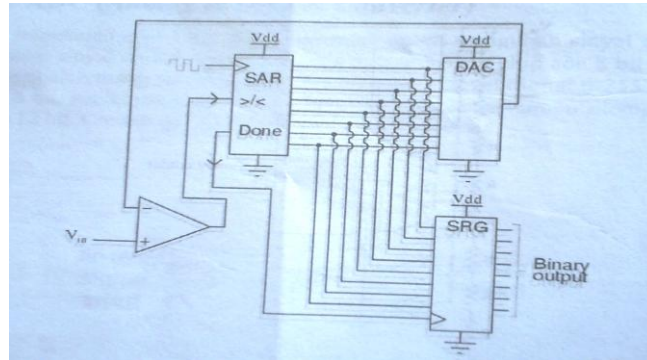
Dikenal sebagai paralel A/D converter sebagai rangkaian ADC yang paling mudah dipelajari. Dibentuk dari sekumpulan komparater yang membandingkan sinyal input ke tegangan referensi menggunakan Op-Amp. Resistor yang digunakan disini harus menggunakan resistor bertoleransi tinggi agar lebih akurat keluaran output dihubungkan ke input dari sebuah prioriti enkoder yang akan menghasilkan output biner diagram berikut merupakan contoh rangkaian flash ADC 3 bit yang umum;



**Gambar 3.7. Rangkaian Flash ADC**

## 2. Successive approximation ADC (SAR)

Rangkaian ini menggunakan Counter yang dikenal sebagai Successive approximation register, yaitu melalui pendekatan berturut-turut untuk mencari nilai yang paling tepat disamping menghitung naik deretan data biner, register ini menghitung dengan mencoba seluruh nilai bit dimulai dari MSB dan diakhiri dengan LSB selama proses perhitungan, register akan memonitor output komparator untuk melihat jika hitungan biner kurang atau lebih besar dari input sinyal analog. SAR dapat mengeluarkan bilangan biner dalam format serial sehingga dapat meniadakan shift register.



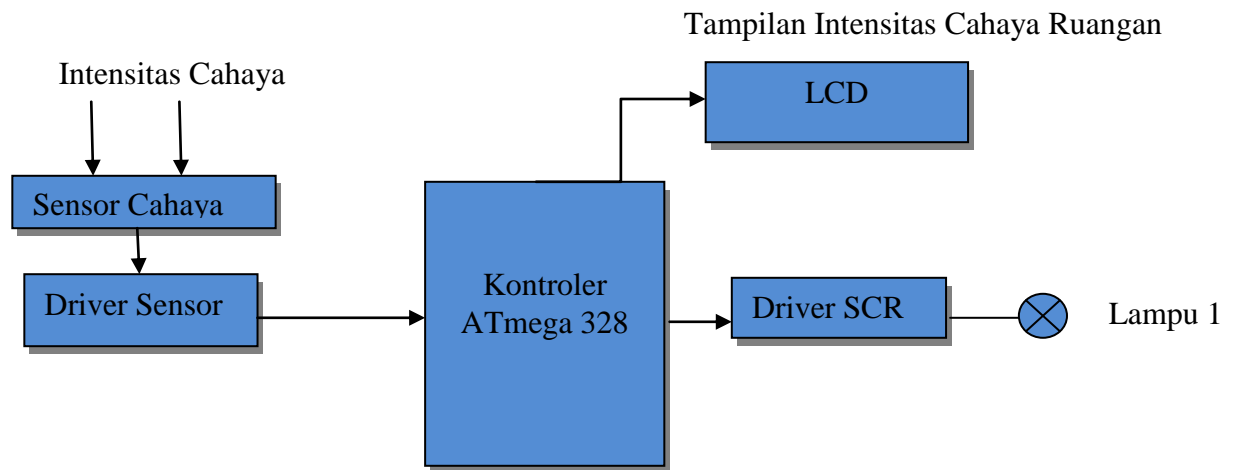
**Gambar 3.8. Rangkaian SAR**

Tegangan referensi dan input ADC adalah 5 volt DC. Dimana tegangan maksimum dari ADC 5 Volt dan keluaran dari ADC 8 bit sehingga :

$$Q = \frac{5}{2^8} = \frac{5}{256} = 0,0195V \dots\dots\dots 3.1$$

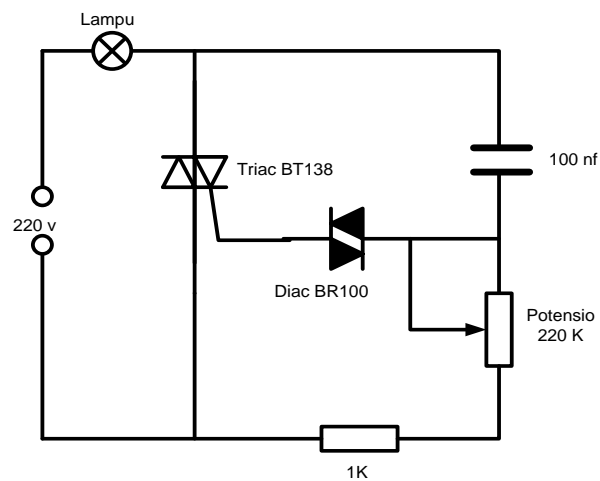
Suatu *Analog to Digital Converter* (ADC) digunakan untuk mengkonversi suatu sinyal analog menjadi suatu sinyal digital yang mewakili besaran sinyal analog tersebut untuk diproses mikrokontroler.

Perancangan sistem Pengendali Penerangan Ruangan meliputi perancangan perangkat lunak (*software*) dan perangkat keras (*hardware*). Sistem yang dirancang akan membentuk suatu sistem pengendali penerangan ruang. Pengendalian penerangan ruang dilakukan dengan mengendalikan besarnya kuantitas cahaya yang dihasilkan oleh lampu. Untuk pengendalian besarnya kuantitas cahaya, maka dilakukan pengaturan tegangan yang diberikan ke lampu. Pada Gambar 3.6. pengaturan tegangan dilakukan oleh blok pengatur tegangan berdasarkan output dari pengendali.



**Gambar 3.9. Blok diagram sistem Pengendalian Penerangan Ruangan**

Adapun salah satu contoh rangkaian sederhana yang digunakan untuk pengontrolan intensitas cahaya ditampilkan pada gambar di bawah ini :

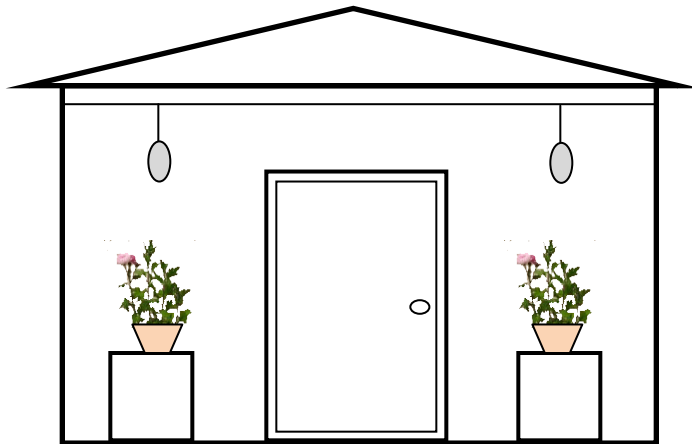


**Gbr 3.10. Rangkaian Pengaturan Daya Lampu Sederhana**

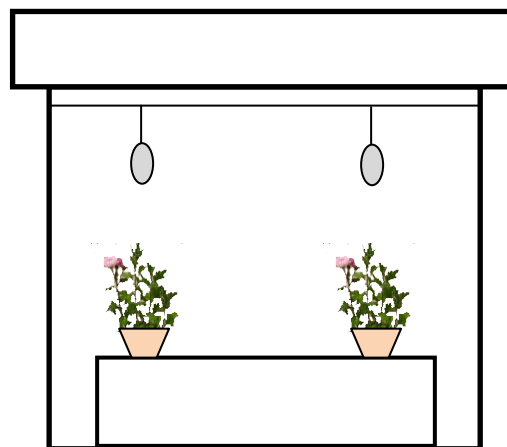


### 3.5. Prototype Sistem

Prototype system yang akan dibuat disini adalah prototype dari Sistem Kendali Intensitas Cahaya Rumah Kaca Cerdas Untuk Budidaya Bunga Krisan. Adapun prototype yang akan di buat seperti pada gambar 3.8.



**Gambar 3.11.(a) Prototype Tampak depan Sistem Kendali Intensitas Cahaya Rumah Kaca Cerdas Untuk Budidaya Bunga Krisan**



**Gambar 3.11.(b) Prototype Tampak Samping Sistem Kendali Intensitas Cahaya Rumah Kaca Cerdas Untuk Budidaya Bunga Krisan**

### **3.6. Skenario Pengujian**

Setelah miniatur rumah kaca selesai dibuat, maka langkah selanjutnya adalah pengambilan data intensitas cahaya dengan alat lux meter. Adapun pengujian yang dilakukan di antaranya :

- ✍ Pengujian ADC
- ✍ Pengujian sensor LDR
- ✍ Pengujian kecerahan rumah kaca
- ✍ Pengujian respon sistem

### **3.7. Teknik Analisa Data**

Adapun Bagian-bagian yang diambil untuk teknik analisis data:

- Pengujian Intensitas cahaya ruangan.
- Pengujian output proses sistem dengan membandingkan data error cahaya dan set\_point.
- Analisis output intensitas cahaya ruangan terhadap pertumbuhan bunga krisan.