

BAB IV

HASIL DAN PEMBAHASAN

Bab ini berisi tentang hasil dan pembahasan sistem kontrol suhu dan pencahayaan pada aquascape yang terdiri dari hasil perancangan dan pengujian sistem kontrol suhu, sistem kontrol HPL menggunakan *smartphone*, konektivitas sensor *bluetooth* dengan Arduino Uno, serta analisis hasil penggunaan alat sistem kontrol suhu dan pencahayaan pada aquascape. Hasil pengujian berupa data-data yang diperoleh untuk memperlihatkan hasil kinerja alat dapat bekerja dengan baik atau tidak. Data-data tersebut dapat dilakukan analisis kinerja perangkat secara keseluruhan berdasarkan pengujian dari masing-masing sistem untuk mendapatkan hasil yang lebih baik dan akurat dan dapat digunakan untuk mengambil kesimpulan akhir terhadap perangkat tersebut.

4.1 Hasil Perancangan *Hardware*

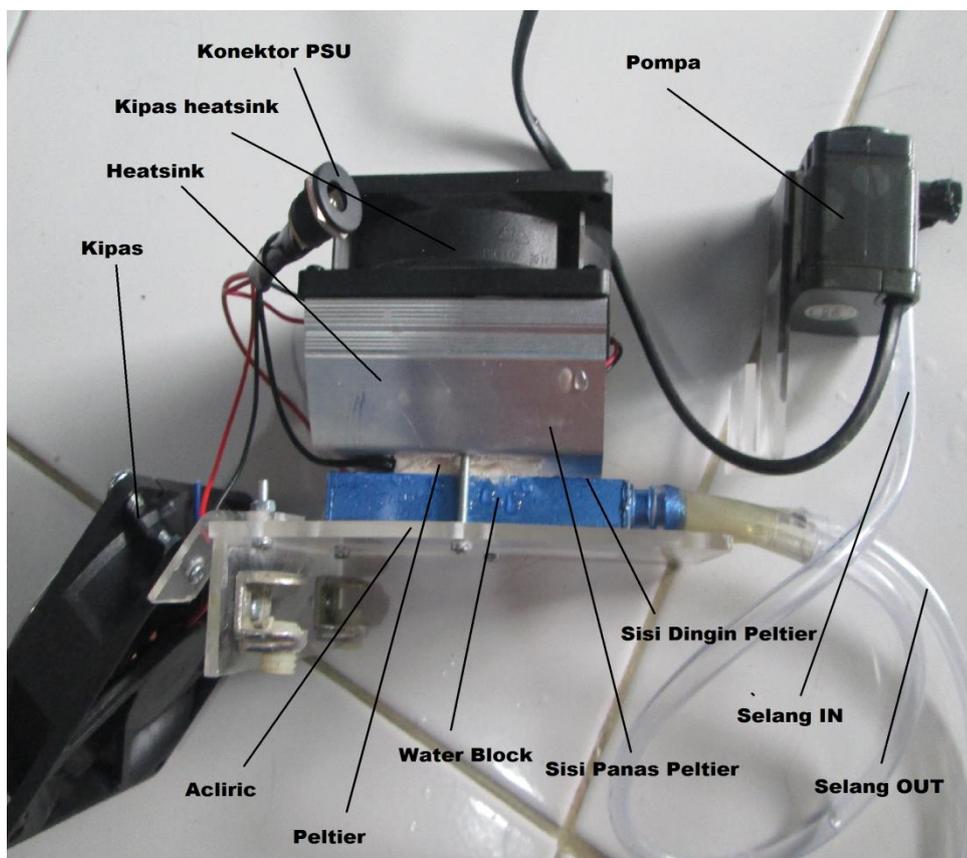
Hasil perancangan *hardware* terdiri dari Arduino Uno dan komponen sistem kontrol suhu dan pencahayaan. Pada sistem kontrol suhu terdiri dari sensor suhu DS18B20, kipas pendingin, peltier, *heatsink*, *waterblock*, pompa mini, Mosfet IRLB 3034, LCD 16x2 dan *powersuply* 12VDC 5A. Sedangkan untuk sistem kontrol pencahayaan terdiri dari HPL (*High power led*), Mosfet IRF540 N, sensor *bluetooth* HC-05, *heatsink*, *powersuply* 12VDC 2A dan *cover* lampu alumunium. Hasil dari rangkaian *hardware* sistem kontrol suhu dan pencahayaan ditunjukkan pada gambar 4.1.



Gambar 4.1 Hasil *Hardware* sistem kontrol suhu dan pencahayaan

4.2 Pengujian Komponen Sistem Kontrol Suhu

Sistem kontrol suhu terdiri dari beberapa komponen antara lain sensor suhu DS18B20, kipas pendingin, peltier, pompa, LCD 16x2, mosfet IRLB 3034 dan Arduino Uno sebagai kontroler. Prinsip kerja dari sistem kontrol suhu ini yaitu sensor DS18B20 akan membaca suhu pada air *aquarium*. Ketika suhu yang terukur mencapai batas nilai yang telah ditentukan pada program arduino yang dibuat, maka arduino akan memberikan perintah pada komponen pendingin dan juga menampilkannya pada LCD. Pada alat ini, batas suhu terendah adalah 26° Celcius, artinya bahwa ketika sensor membaca suhu diatas 26°C maka arduino akan menjalankan perintah agar kipas pendingin dan peltier bekerja untuk mendinginkan suhu air, sebaliknya ketika suhu yang dibatasi telah tercapai yaitu kurang dari atau sama dengan 26° Celcius, maka kipas pendingin dan peltier akan berhenti bekerja. Komponen sistem kontrol suhu ditunjukkan pada gambar 4.2



Gambar 4.2 Komponen Pendingin

4.2.1 Pengujian Sensor DS18B20 dan LCD 16x2

Sensor suhu yang digunakan pada penelitian ini menggunakan modul sensor DS18B20 dilengkapi dengan *waterproof* sehingga dapat melakukan pengukuran suhu didalam air dengan baik. Sensor ini bekerja pada tegangan 5 Volt dan untuk hasil pembacaan sensor akan ditampilkan pada LCD 16x2. Pada gambar 4.3 akan ditampilkan pengujian pengukuran sensor DS18B20 dibandingkan dengan *thermometer* digital dan hasilnya akan ditampilkan pada tabel 4.1



(a) Pengukuran suhu air

(b) Pengukuran suhu ruangan

Gambar 4.3 Hasil pengukuran sensor DS18B20 dan *Thermometer*

Tabel 4.1 Hasil pengujian perbandingan pengukuran menggunakan *thermometer*

| Suhu ruangan pada alat(°C) | Suhu ruangan Pada <i>Thermometer</i> (°C) | Eror (%) | Suhu air Pada alat(°C) | Suhu air Pada <i>Thermometer</i> (°C) | Eror (%) |
|----------------------------|---|----------|------------------------|---------------------------------------|----------|
| 30,81 | 30,80 | 0,03 | 27,12 | 27,10 | 0,07 |
| 31,12 | 31,10 | 0,06 | 28,39 | 28,40 | 0,03 |
| 29,15 | 29,10 | 0,17 | 26,60 | 26,60 | 0 |

Pada gambar 4.3 adalah pengujian dengan mengambil sampel suhu pada ruangan dan suhu pada air. Pengukuran suhu ruangan menggunakan sensor DS18B20 dengan *thermometer* menunjukkan hasil dengan selisih error terbesar yaitu 0,17. Hal ini dikarenakan sensitivitas dan keakuratan tiap sensor berbeda. Dari pengujian ini dikatakan sensor dalam keadaan baik , karena selisih pengukuran sensor dengan alat ukur *thermometer* tidak terlalu jauh.

Untuk melakukan pembacaan sensor dan menampilkannya pada LCD, sensor DS18B20 dikontrol oleh Arduino. Program pada arduino IDE untuk mengakses sensor DS 18B20 akan ditampilkan pada gambar 4.4 dan 4.5.

```
//DS 18B20
#include <OneWire.h> // Library onewire
#include <DallasTemperature.h> // Library untuk mengakses DS18B20
#define DS18B20 10 // Definisikan bahwa PIN 10 adalah DS18B20
OneWire oneWire(DS18B20);
DallasTemperature sensors(&oneWire);
DeviceAddress tempSensor = {0x28, 0xFF, 0x9E, 0x7E, 0x83, 0x17, 0x04, 0x85}; // Alamat ID sensor
char hiSetPoint = 26; // Masukkan nilai batas suhu tertinggi
float tempC;

void setup ()

{
// DS18b20
char alarmTemp;
sensors.begin(); // Memulai membaca nilai sensor
pinMode (kipas, OUTPUT );
sensors.setHighAlarmTemp(tempSensor, hiSetPoint); // Mensetting nilai batas suhu
alarmTemp = sensors.getHighAlarmTemp(tempSensor); |
```

Gambar 4.4 Program IDE Sensor suhu bagian 1

```
//DS18B20
sensors.requestTemperaturesByAddress(tempSensor); // Kirim perintah untuk mendapatkan nilai suhu
if (sensors.hasAlarm(tempSensor)) // Perintah sanggahan

{
lcd.clear();
lcd.setCursor (0,0);
lcd.print ("Temp :");
lcd.print (sensors.getTempC(tempSensor), 2);
lcd.print (" 'C");
lcd.setCursor (0,1);
Serial.println("alarm");
lcd.print("Fan Active");
digitalWrite (kipas, HIGH);
delay(2000);
}

else{
lcd.clear();
lcd.setCursor (0,0);
lcd.print ("Temp :");
lcd.print (sensors.getTempC(tempSensor), 2);
lcd.print (" 'C");
lcd.setCursor (0,1);
Serial.println("no alarm");
lcd.print("Fan NonActive");
digitalWrite (kipas, LOW);
delay(2000);
}
```

Gambar 4.5 Program IDE sensor suhu bagian 2

Pada gambar 4.4 merupakan program awal dimana alamat dan *library* untuk mengakses sensor DS 18B20 perlu dimasukkan. Kemudian mengidentifikasi port Arduino 10 sebagai Pin masukan sensor. Pada bagian ini juga di *setting* bahwa batas suhu tertinggi adalah 26° C. Sehingga pada gambar 4.5 di program akan dilakukan proses bahwa saat suhu mencapai diatas 26°C maka kipas akan aktif dan bila dibawah 26°C kipas akan mati. Untuk mengetahui suhu yang terukur dan status pada kipas dapat dilihat pada LCD seperti pada gambar 4.6.



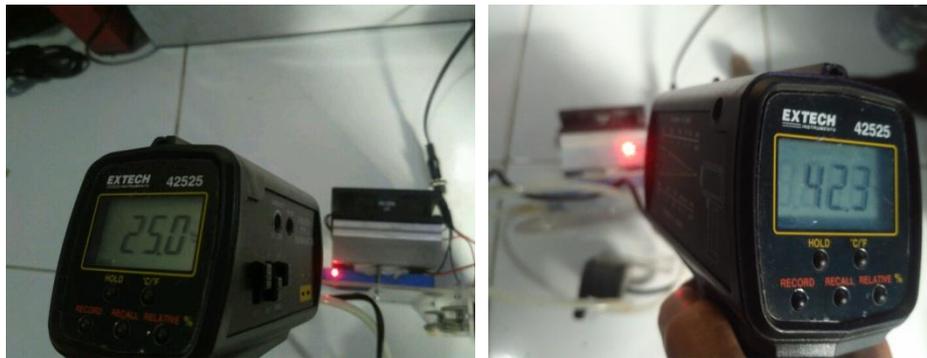
Gambar 4.6 Tampilan pembacaan suhu pada LCD 16x2

4.2.2 Pengujian Peltier dan Kipas Pendingin

Peltier dan kipas pendingin adalah komponen yang digunakan untuk mendinginkan suhu air pada *aquarium*. Peltier dan kipas akan aktif ketika mendapatkan perintah dari Arduino bahwa sensor membaca suhu diatas 26°C. Untuk dapat mendinginkan suhu secara optimal akan dilakukan pengujian terhadap peltier dan kipas pendingin.

Peltier yang digunakan bertipe TEC1-12706 dan akan bekerja optimal pada tegangan 12V 5-8 A. Sedangkan kipas pendingin bekerja pada tegan 12Volt 0,8 A. Ketika *disuply* tegangan maka pada satu sisi peltier akan menjadi panas dan pada sisi sebaliknya menjadi dingin. Karena peltier memiliki 2 sisi yakni *cool side* (*Heat Absorbed*) yang bekerja menyerap kalor (panas) sehingga bagian ini merupakan lempengan yang dingin dan *hot side* (*heat released*) yang bekerja melepas kalor (panas) sehingga bagian ini merupakan lempengan panas. Jadi

semakin dingin *hot side* maka *cold side* akan semakin dingin pula dan bisa sampai dibawah 0 derajat celcius. Untuk meminimalisir suhu pada *Heat side* maka ditambahkan *heatsink* dan kipas pendingin. Pada gambar 4.7 akan ditampilkan pengujian suhu pada sisi-sisi peltier. Hasil pengukuran ditampilkan pada tabel 4.2



(a) Pengujian sisi dingin Peltier (b) Pengujian sisi panas Peltier

Gambar 4.7 Pengujian Peltier

Tabel 4.2 Hasil pengukuran suhu peltier

| Suhu sisi dingin (°C) | Suhu sisi panas (°C) | Selisih suhu sisi panas-dingin (°C) |
|-----------------------|----------------------|-------------------------------------|
| 25,0 | 42,3 | 17,3 |
| 25,2 | 41,1 | 15,9 |
| 26,1 | 44,2 | 18,1 |
| 26,7 | 44,9 | 18,2 |
| 24,9 | 39,8 | 14,9 |
| Nilai rata-rata | | 16,88 |

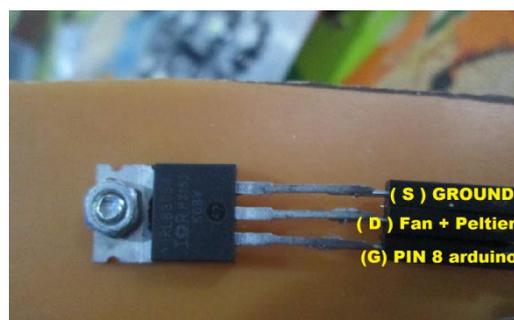
Pada gambar 4.7 adalah pengujian peltier dengan membandingkan suhu antara sisi panas dan sisi dingin peltier. Pengujian dilakukan menggunakan *powersupply* 12V 5A. Setelah beberapa saat pada bagian sisi dingin peltier menjadi dingin dan berembun sedangkan pada bagian sisi lainnya menjadi panas. Pada tabel 4,2 didapat hasil sampel pengukuran menggunakan *thermometer*, dari 5 hasil pengujian didapatkan nilai rata-rata selisih suhu adalah 16,88. Dari pengujian ini

maka dapat ditarik kesimpulan bahwa peltier dalam keadaan normal karena dapat beroperasi dengan baik.

Pada sisi dingin peltier dipasangkan *waterblock* yang berfungsi sebagai media untuk mendinginkan air. Air *aquarium* akan dipompa menuju *waterblock*. *Waterblock* terbuat dari alumunium dan didalamnya terdapat sekat-sekat agar aliran air tidak langsung keluar namun mengalir mengikuti sekat-sekat tersebut dengan tujuan air dapat didinginkan dengan optimal karena sisi *waterblock* terhubung dengan sisi dingin peltier, selain itu *waterblock* terbuat dari alumunium yang berfungsi menyerap dan menjaga kestabilan suhu dingin dari peltier.

4.2.3 Pengujian Mosfet IRLB 3034

Pada alat sistem kontrol suhu ini, mosfet IRLB 3034 berperan sebagai saklar maka hanya menggunakan mosfet pada kondisi saturasi (*ON*) dan kondisi *cut-off* (*OFF*). Mosfet memiliki impedansi masukan (*gate*) sangat tinggi sehingga dengan menggunakan mosfet sebagai saklar *elektronik*, memungkinkan untuk menghubungkannya dengan semua jenis gerbang logika. Dengan menjadikan mosfet sebagai saklar, maka dapat digunakan untuk mengendalikan beban dengan arus yang tinggi. Mosfet IRLB 3034 dapat beroperasi pada tegangan maksimal 40V 120A. Konfigurasi terminal kaki mosfet IRLB 3034 dapat dilihat pada gambar 4.8



Gambar 4.8 Konfigurasi terminal mosfet IRLB 3034

Pengujian pada mosfet dapat dilihat pada gambar 4.9. Mosfet memiliki 3 kaki yaitu *Gate*, *Drain*, dan *Source*. Pada *Gate* dihubungkan dengan PIN arduino 8 karena pada *gate* inilah nanti akan dikirimkan sinyal tegangan agar mengalir

pada *ground* sedangkan pada kaki *Drain* sebagai output. Pada pengujian ini untuk mengetahui ada tidaknya tegangan pada *gate* ketika pin 8 pada arduino diberikan masukan *HIGH*.



(a) Saat diberi masukan *HIGH*



(b) Saat diberi masukan *LOW*

Gambar 4.9 Pengujian mosfet IRLB3034

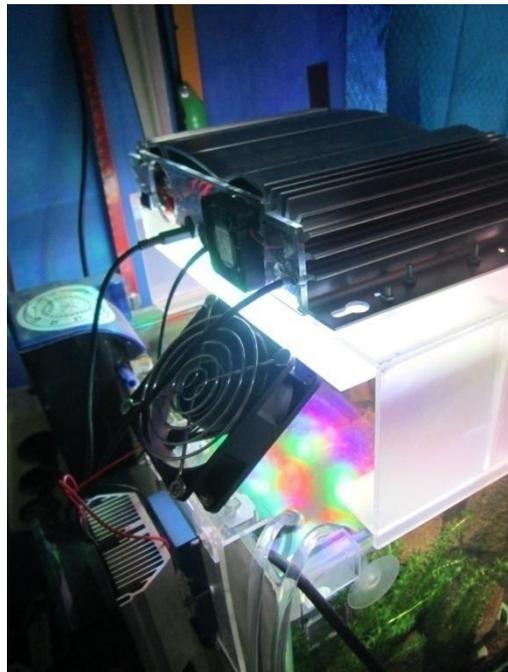
Pada gambar (a) 4.9 adalah ketika *gate* mosfet mendapatkan inputan tegangan pada pin 8 arduino, sehingga LED putih sebagai indikator akan menyala karena mosfet menjadi *ON* dan arus dapat mengalir. Sebaliknya, pada gambar (b) ketika *gate* tidak diberi tegangan maka mosfet dalam keadaan *OFF* karena tidak ada arus yang mengalir. Hasil pengukuran (a) menunjukkan tegangan saat *gate* diberi inputan yaitu 4,95 Volt sedangkan pada gambar (b) saat tidak diberi masukan yaitu 0 Volt. Nilai tegangan pada gambar (a) merupakan nilai maksimal karena tegangan keluaran pada arduino sebesar 5 Volt. Program pada Arduino IDE untuk memberikan perintah ditunjukkan pada gambar 4.10.

```
{
//DS18B20
sensors.requestTemperaturesByAddress(tempSensor); // Kirim perintah untuk mendapatkan nilai suhu
if (sensors.hasAlarm(tempSensor)) // Perintah sanggahan
// Ketika sensor membaca suhu diatas 26' maka
{lcd.clear();
lcd.setCursor (0,0);
lcd.print ("Temp :");
lcd.print (sensors.getTempC(tempSensor), 2);
lcd.print (" 'C");
lcd.setCursor (0,1);
Serial.println("alarm");
lcd.print("Fan Active");
digitalWrite (cooler, HIGH);
delay (2000);
}
else{
lcd.clear();
lcd.setCursor (0,0);
lcd.print ("Temp :");
lcd.print (sensors.getTempC(tempSensor), 2);
lcd.print (" 'C");
lcd.setCursor (0,1);
Serial.println ("no alarm");
lcd.print("Fan NonActive");
digitalWrite (cooler, LOW);
delay (2000);
}
```

Gambar 4.10 Program Arduino IDE kontrol mosfet IRLB3034

4.3 Pengujian Sistem Kontrol Suhu

Pada pengujian ini adalah untuk mengetahui hasil kinerja alat sistem kontrol suhu pada aquascape apakah dapat bekerja dengan optimal atau tidak. Pengujian dilakukan pada *aquarium* dengan ukuran $p = 30\text{cm}$ $l = 20\text{cm}$ $t = 25\text{cm}$. Waktu dilakukanya pengamatan adalah 24 jam. Pada gambar 4.11 adalah *aquarium* yang telah dipasang sistem kontrol suhu. Hasil dari pengamatan akan ditunjukkan pada tabel 4.3 dan 4.4.



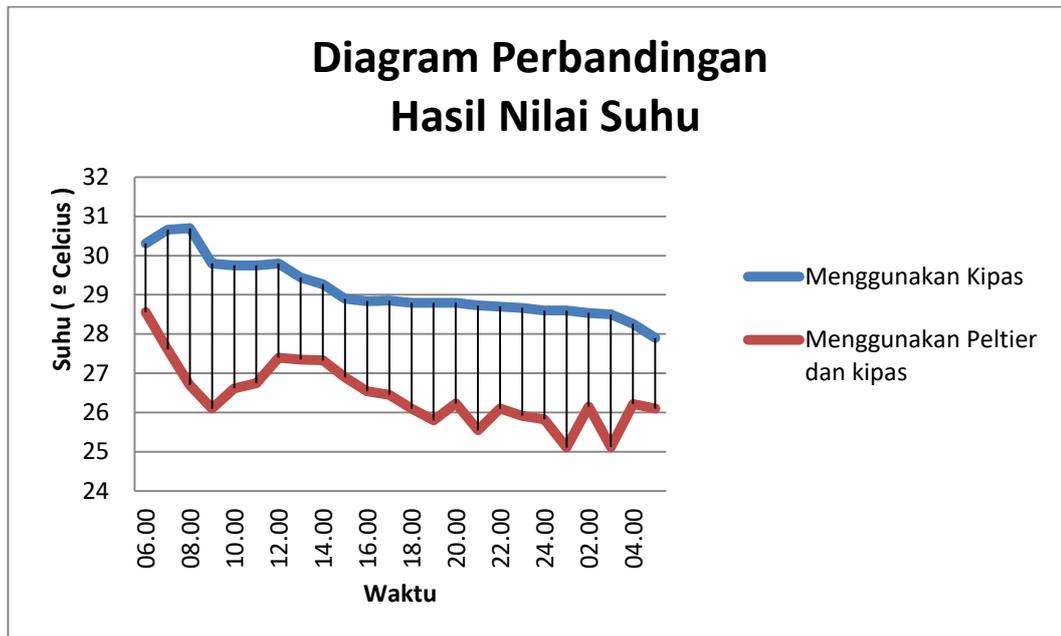
Gambar 4.11 Pengujian sistem kontrol suhu

Pada tabel 4.3 adalah hasil perbandingan ketika dilakukan pengujian selama 24 jam dengan membandingkan antara pendinginan menggunakan kipas dengan pendinginan menggunakan peltier dan kipas. Hasil tersebut kemudian akan ditampilkan menggunakan diagram garis pada gambar 4.12 untuk melihat hasil perbedaan dari pengujian tersebut.

Tabel 4.3 Hasil perbandingan pengukuran suhu

| No | Waktu | Suhu (°C) | | Suhu (°C) | |
|----|-------|-----------|------------------------|-----------|------------------------------------|
| | | Ruangan | Saat menggunakan kipas | Ruangan | Saat menggunakan Peltier dan kipas |
| 1 | 06.00 | 30,37 | 30,31 | 30,35 | 28,56 |
| 2 | 07.00 | 30,92 | 30,66 | 30,8 | 27,61 |
| 3 | 08.00 | 30,85 | 30,7 | 30,95 | 26,7 |
| 4 | 09.00 | 31,5 | 29,8 | 31,8 | 26,1 |
| 5 | 10.00 | 32,4 | 29,75 | 32,1 | 26,62 |
| 6 | 11.00 | 33,6 | 29,75 | 33,5 | 26,75 |
| 7 | 12.00 | 33,7 | 29,8 | 33,8 | 27,4 |
| 8 | 13.00 | 32,75 | 29,44 | 32,75 | 27,35 |
| 9 | 14.00 | 32,7 | 29,27 | 32,7 | 27,33 |
| 10 | 15.00 | 32,6 | 28,9 | 32,5 | 26,9 |
| 11 | 16.00 | 32,2 | 28,84 | 32,1 | 26,54 |
| 12 | 17.00 | 32 | 28,85 | 32 | 26,45 |
| 13 | 18.00 | 31,05 | 28,8 | 31,15 | 26,1 |
| 14 | 19.00 | 31,15 | 28,8 | 31,15 | 25,8 |
| 15 | 20.00 | 31,1 | 28,8 | 31,1 | 26,23 |
| 16 | 21.00 | 31,15 | 28,73 | 31,15 | 25,55 |
| 17 | 22.00 | 31,05 | 28,7 | 31,15 | 26,1 |
| 18 | 23.00 | 31,05 | 28,67 | 31,05 | 25,92 |
| 19 | 24.00 | 30,7 | 28,6 | 30,8 | 25,83 |
| 20 | 01.00 | 30,8 | 28,6 | 30,8 | 25,11 |
| 21 | 02.00 | 30,65 | 28,54 | 30,55 | 26,15 |
| 22 | 03.00 | 30,5 | 28,5 | 30,6 | 25,12 |
| 23 | 04.00 | 30,55 | 28,26 | 30,55 | 26,22 |
| 24 | 05.00 | 30,55 | 27,9 | 30,65 | 26,1 |

Dari tabel 4.3 dapat dianalisis bahwa hasil pengukuran suhu terendah yang didapatkan dengan menggunakan pendingin berupa kipas adalah sebesar 27,90° Celcius, sedangkan apabila menggunakan peltier dan kipas pendingin suhu terendah yang didapatkan adalah sebesar 25,11° Celcius. Pada gambar 4.12 akan ditampilkan diagram garis hasil perbandingan tersebut.



Gambar 4.12 Diagram Perbandingan Hasil Nilai Suhu

Dari hasil pengukuran suhu pada gambar 4.12, dapat dianalisis bahwa terdapat perbedaan nilai suhu berdasarkan penggunaan tipe pendingin. Padagaris berwarna biru adalah nilai yang didapat ketika menggunakan pendingin berupa kipas. Dalam waktu 24 jam pengujian, didapatkan hasil pengukuran suhu terendah sebesar 27,90 dan suhu cenderung lebih stabil pada kisaran suhu 28-29° Celcius. Sedangkan pada garis merah, yaitu ketika menggunakan peltier dan kipas, suhu terendah yang didapatkan adalah 25,11° Celcius dan suhu stabil pada kisaran 25-27° Celcius. Sehingga pada pengujian ini didapatkan kesimpulan bahwasanya penggunaan pendingin peltier dan kipas akan lebih efektif dari pada hanya menggunakan kipas, karena saat menggunakan peltier dan kipas, mampu menurunkan suhu hingga batas yang diharapkan yaitu pada 26°Celcius. Sedangkan penggunaan pendingin yang hanya menggunakan kipas tidak mampu menurunkan suhu hingga 26° Celcius.

Tabel 4.4 Hasil pengujian sistem kontrol suhu

| No | Waktu | Suhu (°C) | | | Error (%) | Status Pendingin |
|----|-------|-----------|-----------------|--------------------|-----------|------------------|
| | | Ruangan | <i>Aquarium</i> | <i>Thermometer</i> | | |
| 1 | 06.00 | 30,35 | 28,56 | 28,7 | 0,49 | Active |
| 2 | 07.00 | 30,8 | 27,61 | 27,5 | 0,39 | Active |
| 3 | 08.00 | 30,95 | 26,7 | 26,7 | 0 | Active |
| 4 | 09.00 | 31,8 | 26,1 | 26,1 | 0 | Active |
| 5 | 10.00 | 32,1 | 26,62 | 26,6 | 0,08 | Active |
| 6 | 11.00 | 33,5 | 26,75 | 26,6 | 0,56 | Active |
| 7 | 12.00 | 33,8 | 27,4 | 27,4 | 0 | Active |
| 8 | 13.00 | 32,75 | 27,35 | 27,3 | 0,04 | Active |
| 9 | 14.00 | 32,7 | 27,33 | 27,3 | 0,18 | Active |
| 10 | 15.00 | 32,5 | 26,9 | 26,9 | 0 | Active |
| 11 | 16.00 | 32,1 | 26,54 | 26,5 | 0,15 | Active |
| 12 | 17.00 | 32 | 26,45 | 26,4 | 0,18 | Active |
| 13 | 18.00 | 31,15 | 26,1 | 26,1 | 0 | Active |
| 14 | 19.00 | 31,15 | 25,8 | 25,8 | 0 | Non Active |
| 15 | 20.00 | 31,1 | 26,23 | 26,2 | 0,26 | Active |
| 16 | 21.00 | 31,15 | 25,55 | 25,5 | 0,19 | Non Active |
| 17 | 22.00 | 31,15 | 26,1 | 26,1 | 0 | Active |
| 18 | 23.00 | 31,05 | 25,92 | 25,9 | 0,07 | Non Active |
| 19 | 24.00 | 30,8 | 25,83 | 25,8 | 0,12 | Non Active |
| 20 | 01.00 | 30,8 | 25,11 | 25,1 | 0,03 | Non Active |
| 21 | 02.00 | 30,55 | 26,15 | 26,1 | 0,15 | Active |
| 22 | 03.00 | 30,6 | 25,12 | 25,1 | 0,07 | Non Active |
| 23 | 04.00 | 30,55 | 26,22 | 26,1 | 0,45 | Active |
| 24 | 05.00 | 30,65 | 26,1 | 26,1 | 0 | Active |

Dari hasil pengamatan pada tabel 4.4 maka dapat dianalisis bahwa suhu tertinggi didapat pada pukul 06.00 yaitu 28,50° C dan suhu terendah pada pukul 03.00 dengan suhu 25,10°C. Suhu tertinggi tersebut dikarenakan awal dari pendingin beroperasi sehingga proses pendinginan belum maksimal dan membutuhkan waktu. Setelah beberapa jam mulai terlihat bahwa terdapat penurunan suhu hingga stabil pada kisaran 26° C. Dalam pengujian ini terdapat beberapa faktor yang mempengaruhi temperatur air *aquarium*, antara lain yaitu

suhu ruangan sangat berpengaruh terhadap cepat dan lambatnya proses pendinginan, kemudian lampu yang menyala pada pukul 16.00-23.00 juga berpengaruh terhadap perubahan suhu karena dari nyala lampu juga menghasilkan panas.

Nilai data eror didapat menggunakan rumus:

$$\frac{(SuhuTerbaca - SuhuSebenarnya)}{SuhuTerbaca} \times 100\%$$

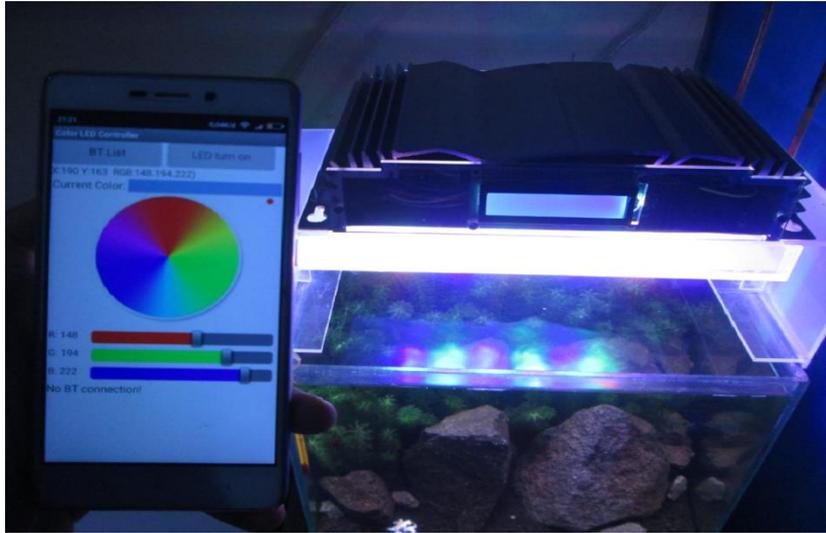
$$\text{Contoh : } \frac{(28,56 - 28,7)}{28,56} \times 100\% = 0,49 \%$$

Pada tabel 4.4 didapat hasil nilai eror tertinggi adalah 0,56 % dan eror terendah adalah 0. Perbedaan tersebut karena faktor sensitifitas serta keakuratan masing – masing sensor berbeda.

Dari hasil pengujian sistem kontrol suhu yang telah dilakukan dapat ditarik kesimpulan bahwasanya alat dapat beroperasi dengan baik dan mendapatkan hasil yang diinginkan karena dapat mengatasi permasalahan temperatur suhu yang tinggi. Sedangkan setelah dipasangnya alat, suhu menjadi lebih rendah dan stabil pada kisaran 25-26° C.

4.4 Pengujian Komponen Sistem Kontrol Pencahayaan

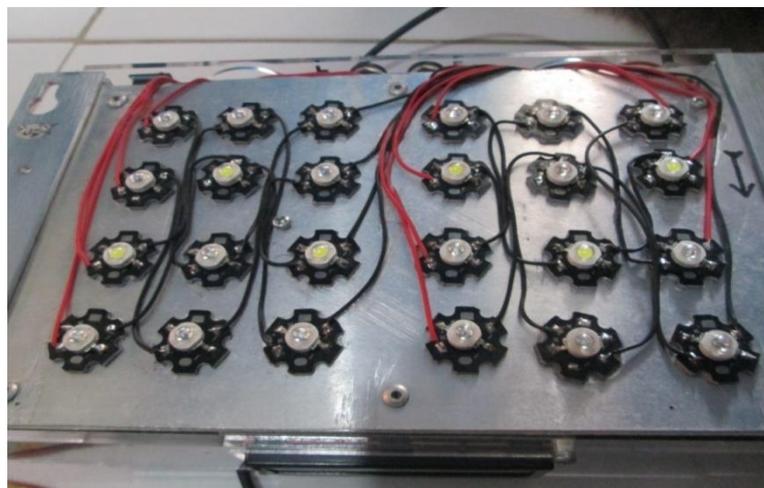
Komponen sistem kontrol pencahayaan terdiri dari HPL (*High power led*) , Mosfet IRF 540N, Sensor *Bluetooth* HC-05, dan Aplikasi *BT_LED_Controller* pada *smartphone* . Prinsip kerja dari sistem kontrol pencahayaan ini yaitu agar pengguna lebih mudah mengontrol dan memilih warna untuk pencahayaan pada *aquarium*. Pengontrolan dilakukan menggunakan aplikasi pada *smartphone*. Untuk mengkoneksikan *smartphone* dengan arduino digunakan sensor *bluetooth* HC-05. Sistem kontrol pencahayaan pada *aquascape* akan ditampilkan pada gambar 4.13.



Gambar 4.13 Komponen Sistem Kontrol Pencahayaan

4.4.1 Pengujian HPL (*High power led*)

Pada sistem kontrol pencahayaan untuk aquascape, HPL merupakan komponen yang berfungsi sebagai sumber cahaya. Oleh karena itu pengujian HPL dilakukan untuk mengetahui kinerja apakah dapat beroperasi dengan baik atau tidak. HPL memproduksi intensitas cahaya lampu yang lebih kuat, atau bisa disebut yang paling kuat di antara semua jenis lampu LED. Namun, lampu LED ini memiliki potensi untuk lebih cepat panas dibandingkan LED lain sehingga perlu dipasangkan *heatsink* untuk memperpanjang umur komponen HPL tersebut. Pada gambar 4.14 akan ditampilkan hasil perakitan HPL.



Gambar 4.14 Hasil perakitan HPL

Pada gambar 4.14 merupakan hasil perakitan HPL yang dirangkai secara seri dan paralel. HPL yang digunakan dapat bekerja secara optimal pada tegangan 3-3,5 V dan arus 350mA. HPL yang digunakan memiliki daya 1 watt dengan warna merah 6 buah, hijau 6 buah, dan biru 6 buah. Total daya dari HPL yang digunakan adalah 18 watt.

Sumber yang digunakan untuk HPL menggunakan *powersupply* 12V2A. HPL dirangkai secara seri dan paralel agar dapat diatur dengan mudah sesuai dengan pengelompokan warnanya. Perhitungan rangkain HPL seperti berikut :

$$\text{PowerSupply (Vs)} = 12\text{V } 2\text{A}$$

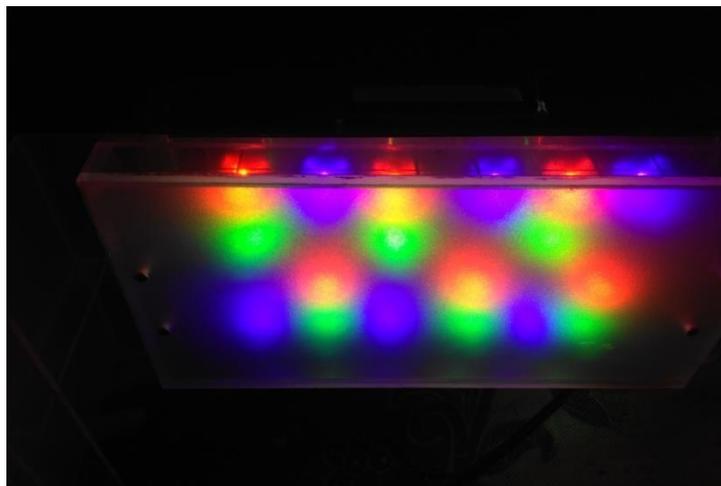
$$\text{Tegangan HPL (Vf)} = 3,3 \text{ V}$$

$$\text{Arus HPL (If)} = 350\text{mA} = 0,35\text{A}$$

Maka dari tegangan sumber kita bagi terlebih dahulu untuk memperkirakan jumlah HPL yang dirangkai seri yaitu $12\text{V} : 3,3 = 3,6$. Maka didapat nilai bahwa rangkaian dapat diseri sebanyak 3 buah HPL. Kemudian ditambahkan resistor agar komponen dapat bekerja optimal dan tidak mudah rusak.

$$R = \frac{(Vs-Vf)}{If} = \frac{(12-(3,3 \times 3 \text{ seri}))}{0,35} = \frac{(12-9,9)}{0,35} = 6 \text{ Ohm } 1 \text{ Watt}$$

Dari hasil tersebut bahwa pada sumber 12V2A, HPL dapat dirangkai secara seri sebanyak 3 buah dan paralel sebanyak 6 buah. Hasil dari nyala HPL dapat dilihat pada gambar 4.15.



Gambar 4.15 Hasil nyala HPL

4.4.2 Pengujian Mosfet IRF 540 N

Mosfet IRF540 N merupakan komponen yang digunakan sebagai PWM untuk melakukan terang redup (*dimmer*) pada HPL. Oleh karena itu pada kaki *gate* mosfet dihubungkan dengan pin keluaran analog arduino yang dapat digunakan untuk pengontrolan PWM. PWM (*Pulse Width Metode*) memanipulasi keluaran digital sedemikian rupa sehingga menghasilkan sinyal analog yang dapat diatur. Variasi nilai output analog dinyatakan pada kondisi *HIGH* untuk puncaknya (5 Volt) dan *LOW* (0 Volt) untuk nilai terendahnya. Untuk mengatur nilai PWM pada program menggunakan fungsi `analogWrite ([NomorPin] , [nilai])` ;seperti *contoh* program pada gambar 4.16.

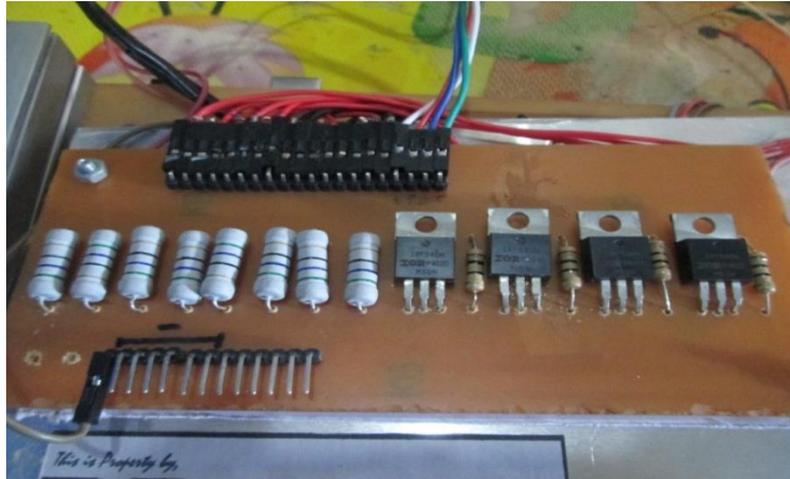
```
const int PIN_5 = 5;

void setup() {
  // put your setup code here, to run once:
  pinMode ( PIN_5 , OUTPUT );
}

void loop() {
  // put your main code here, to run repeatedly:
  analogWrite ( PIN_5, 255 ;
  delay (1000);
  analogWrite ( PIN_5, 150 ;
  delay (1000);
  analogWrite ( PIN_5, 0];
  delay (1000);
```

Gambar 4.16 Penggunaan fungsi *analogWrite* pada IDE

Nilai parameter pada PWM arduino berkisar antara 0 (*Low*) hingga 255 (*High*). Sebagai contoh saat ingin mengatur pada 0% maka diatur nilai parameter pada 0, dan bila ingin mengatur nilai parameter 100% maka diatur pada nilai 255. Jadi , semisal akan mengatur pada 50% maka nilai yang harus diatur adalah 127 (50% x 255). Keluaran PWM dapat diuji dengan mengukur tegangan *gate* mosfet dan tegangan HPL serta melihat dari nyala terang dan redupnya cahaya yang dihsilkan HPL untuk menguji apakah mosfet yang digunakan dapat bekerja secara optimal atau tidak. Hasil dari rangkain mosfet ditunjukkan pada gambar 4.17 dan hasil data pengukuran pengujian mosfet pada tabel 4.5.



Gambar 4.17 Rangkaian mosfet sistem kontrol pencahayaan

Tabel 4.5 Pengujian Mosfet IRF540 N

(a) Pengujian Mosfet IRF540N *channel*merah

| No | Set (%) | Nilai Parameter PWM | V. HPL (V) | I.HPL (A) | V. Gate Terukur (V) | V.Gate sebenarnya (V) | Error (%) | LUX (candela) |
|----|---------|---------------------|------------|-----------|---------------------|-----------------------|-----------|---------------|
| 1 | 0 | 0 | 0,01 | 0 | 0 | 0 | 0 | 0 |
| 2 | 10 | 26 | 1,82 | 31 | 0,53 | 0,5 | 5,66 | 24 |
| 3 | 20 | 51 | 2,05 | 64 | 1,03 | 1 | 2,91 | 43 |
| 4 | 30 | 77 | 2,25 | 92 | 1,57 | 1,5 | 4,45 | 65 |
| 5 | 40 | 102 | 2,44 | 123 | 1,99 | 2 | 0,5 | 88 |
| 6 | 50 | 127 | 2,61 | 156 | 2,48 | 2,5 | 0,8 | 112 |
| 7 | 60 | 153 | 2,77 | 188 | 2,98 | 3 | 0,67 | 135 |
| 8 | 70 | 178 | 2,9 | 218 | 3,46 | 3,5 | 1,15 | 155 |
| 9 | 80 | 204 | 3,05 | 247 | 3,96 | 4 | 1,01 | 180 |
| 10 | 90 | 230 | 3,17 | 279 | 4,47 | 4,5 | 0,67 | 201 |
| 11 | 100 | 255 | 3,23 | 310 | 4,97 | 5 | 0,64 | 224 |

(b) Pengujian Mosfet IRF540N *channel*hijau

| No | Set (%) | Nilai Parameter PWM | V. HPL (V) | I.HPL (A) | V. Gate Terukur (V) | V.Gate sebenarnya (V) | Error (%) | LUX (candela) |
|----|---------|---------------------|------------|-----------|---------------------|-----------------------|-----------|---------------|
| 1 | 0 | 0 | 0,01 | 0 | 0 | 0 | 0 | 0 |
| 2 | 10 | 26 | 1,73 | 31 | 0,48 | 0,5 | 4,16 | 44 |
| 3 | 20 | 51 | 2 | 63 | 1,03 | 1 | 2,91 | 92 |
| 4 | 30 | 77 | 2,21 | 97 | 1,58 | 1,5 | 5,06 | 135 |
| 5 | 40 | 102 | 2,4 | 129 | 1,95 | 2 | 2,56 | 180 |

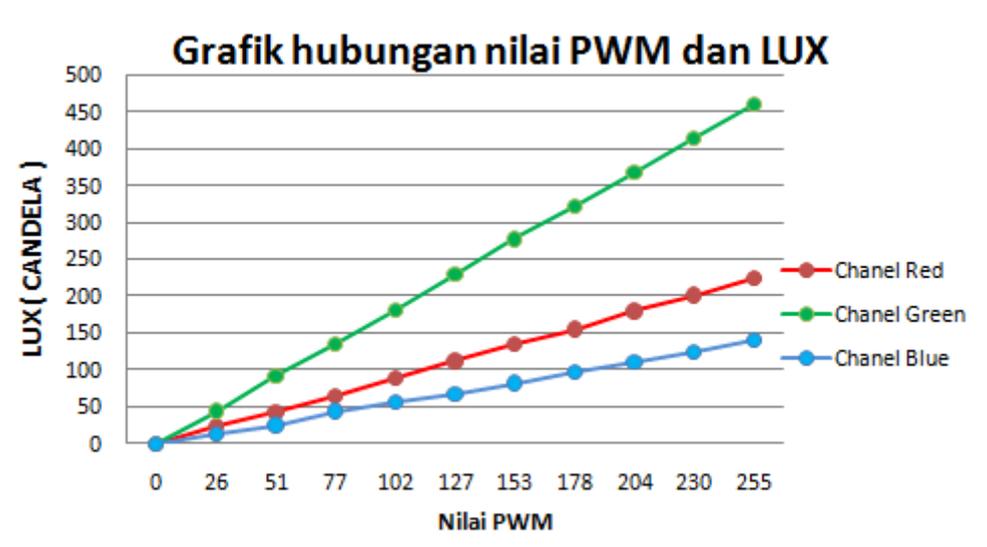
| | | | | | | | | |
|----|-----|-----|------|-----|------|-----|------|-----|
| 6 | 50 | 127 | 2,55 | 161 | 2,48 | 2,5 | 0,8 | 230 |
| 7 | 60 | 153 | 2,73 | 193 | 2,9 | 3 | 3,44 | 277 |
| 8 | 70 | 178 | 2,81 | 224 | 3,48 | 3,5 | 0,57 | 322 |
| 9 | 80 | 204 | 2,9 | 255 | 3,88 | 4 | 3,09 | 367 |
| 10 | 90 | 230 | 3,05 | 288 | 4,47 | 4,5 | 0,67 | 413 |
| 11 | 100 | 255 | 3,13 | 321 | 4,99 | 5 | 0,21 | 459 |

(c) Pengujian Mosfet IRF540N *chanel*biru

| No | Set (%) | Nilai Parameter PWM | V. HPL (V) | I.HPL (A) | V. Gate Terukur (V) | V.Gate sebenarnya (V) | Error (%) | LUX (candela) |
|----|---------|---------------------|------------|-----------|---------------------|-----------------------|-----------|---------------|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 10 | 26 | 1,71 | 34 | 0,49 | 0,5 | 2,04 | 13 |
| 3 | 20 | 51 | 2,2 | 67 | 1 | 1 | 0 | 25 |
| 4 | 30 | 77 | 2,29 | 98 | 1,55 | 1,5 | 3,22 | 44 |
| 5 | 40 | 102 | 2,44 | 132 | 1,98 | 2 | 1,01 | 56 |
| 6 | 50 | 127 | 2,58 | 167 | 2,49 | 2,5 | 0,38 | 68 |
| 7 | 60 | 153 | 2,78 | 199 | 2,98 | 3 | 0,67 | 82 |
| 8 | 70 | 178 | 2,9 | 234 | 3,49 | 3,5 | 0,28 | 98 |
| 9 | 80 | 204 | 2,97 | 264 | 3,87 | 4 | 3,35 | 111 |
| 10 | 90 | 230 | 3,35 | 299 | 4,48 | 4,5 | 0,44 | 125 |
| 11 | 100 | 255 | 3,48 | 330 | 4,97 | 5 | 0,6 | 140 |

Dari tabel 4.5 didapatkan hasil pengukuran dari 3 warna HPL yaitu merah, biru, dan hijau. Masing-masing warna mendapatkan hasil yang berbeda. Dari ketiga pengujian tersebut didapatkan nilai eror yang kecil.

Dari data tersebut akan ditampilkan pada grafik hubungan antara nilai parameter dari PWM dengan intensitas cahaya pada gambar 4.18. Pada gambar 4.18 dapat dianalisis bahwa pada ketiga *chanel* warna lampu akan semakin bertambah nilai LUX nya apabila nilai PWM nya juga bertambah. Artinya, lampu akan semakin terang ketika nilai PWM semakin besar, sebaliknya lampu akan semakin redup ketika nilai PWM semakin kecil.

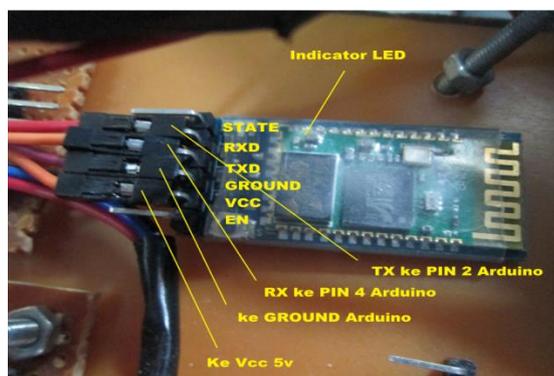


Gambar 4.18 Grafik hubungan nilai PWM dengan LUX

Dari pengujian mosfet tersebut dikatakan bahwa mosfet IRF540N dalam kondisi yang baik dan dapat bekerja secara optimal untuk melakukan proses *dimmer*.

4.4.3 Pengujian Konektifitas

Pada sistem kontrol pencahayaan ini, pengontrolan cahaya dan pemilihan warna dilakukan melalui aplikasi pada *smartphone*. Untuk dapat terkoneksi antara *smartphone* dengan arduino digunakan sensor *Bluetooth* HC-05. Pengaplikasian sensor ini untuk mempermudah pengguna untuk melakukan pengontrolan dengan komunikasi nirkabel atau wireless sehingga dapat melakukan monitoring ataupun pengontrolan. Pada gambar 4.19 adalah rangkaian dari sensor HC-05.



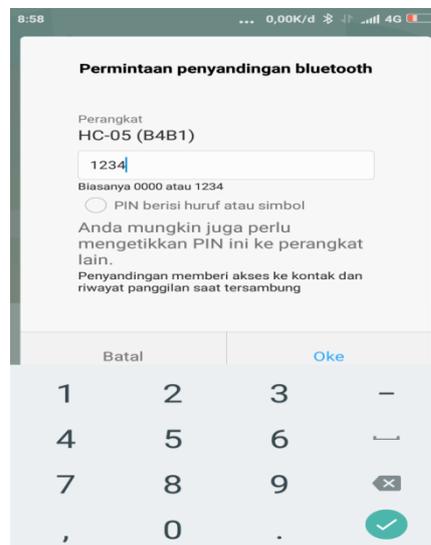
Gambar 4.19 Sensor HC-05

^ Sensor HC-05 bekerja pada tegangan 3,3 – 6 Volt dan memiliki jangkauan jarak efektif pada *range* 10 meter. Untuk mengakses sensor HC-05 pada program IDE diperlukan *library* seperti pada gambar 4.20.

```
#include <SoftwareSerial.h>
#include <Wire.h> //ini library
SoftwareSerial BT(2,4); //definisikan pin tx rx
|
void setup ()
{
BT.begin(9600); //My HC-05 module default baud rate is 9600
```

Gambar 4.20 Program pada IDE mengakses HC-05

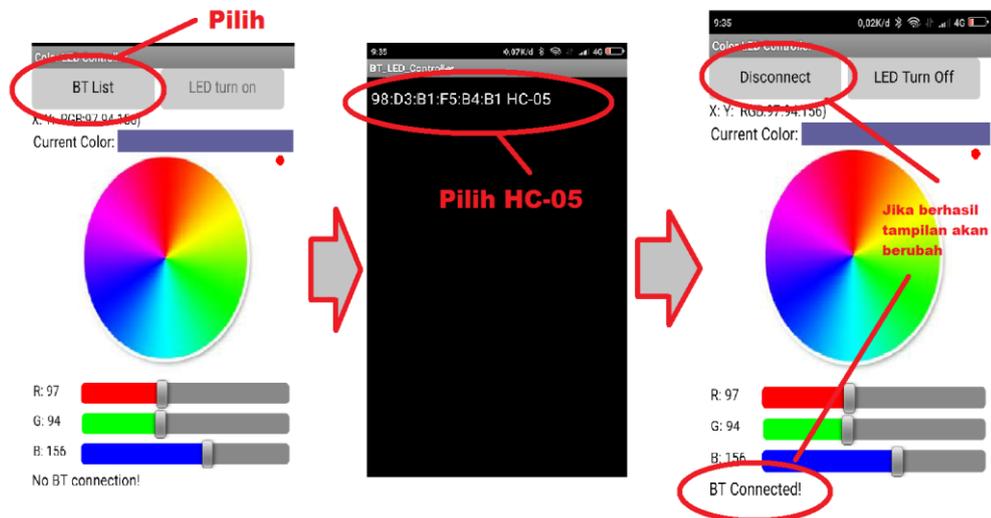
Untuk mengetahui apakah sensor HC-05 dapat terkoneksi dengan *smartphone* maka perlu diuji dengan menyandingkan koneksi *bluetooth* seperti pada gambar 4.21.



Gambar 4.21 Proses penyandingan *bluetooth*

Ketika sandi yang dipasangkan sudah benar, maka otomatis *smartphone* telah mendapatkan akses untuk dapat terhubung dengan sensor HC-05. Untuk mengetahui berhasil atau tidaknya, pada daftar perangkat *bluetooth* akan muncul perangkat HC-05 jika proses penyandingan berhasil. Kemudian untuk dapat

terkoneksi pada aplikasi *LED_Controller* maka sensor HC-05 perlu terkoneksi. Pada gambar 4.22 merupakan tampilan aplikasi saat mengkoneksikan HC-05.



Gambar 4.22 Tampilan proses koneksi HC-05 pada aplikasi *LED_Controller*

Pada aplikasi *LED_Controller* terdapat beberapa tampilan menu untuk mengatur pencahayaan dan tampilan untuk memonitoring, sehingga pengguna dapat lebih mudah dalam melakukan pengontrolan dan memilih cahaya sesuai yang diinginkan. Pada gambar 4.23 akan ditampilkan program pada Arduino IDE untuk mengakses dan melakukan proses pengontrolan pada aplikasi *BT_LED_Controller*.

```
//RGB Led pin
int PIN_RED = 3;
int PIN_GREEN = 5;
int PIN_BLUE = 6;
int PIN_WHITE = 9;
//RED LED at Pin 13
int RED_LED = 13;
String RGB = ""; //Ruang RGB dari BT
String RGB_Previous = "255.255.255"; //setting default
String ON = "ON"; //cek jika menerima perintah on
String OFF = "OFF"; //cek jika menerima perintah off
boolean RGB_Completed = false;
```

(a) Bagian satu

```

while(BT.available()){
char ReadChar = (char)BT.read();

    // tanda kurung tutup menyatakan penyelesaian string
    if(ReadChar == '\n'){
        RGB_Completed = true;
    }else{
        RGB += ReadChar;
    }
}

// Ketika kode perintah diterima sepenuhnya dengan karakter akhir '\n'
if(RGB_Completed){
//print out ke window
    Serial.print("RGB:");
    Serial.print(RGB);
    Serial.print("    PreRGB:");
    Serial.println(RGB_Previous);

    if(RGB==ON){
        digitalWrite(13,HIGH);
        RGB = RGB_Previous; //LED 13 di on kan
        Light_RGB_LED();

    }else if(RGB==OFF){
        digitalWrite(13,LOW);
        RGB = "0.0.0"; //kirim string off
        Light_RGB_LED();
    }else{

```

(b) Bagian dua

```

void Light_RGB_LED(){

    int SP1 = RGB.indexOf('.');
    int SP2 = RGB.indexOf('.', SP1+1);
    int SP3 = RGB.indexOf('.', SP2+1);
    String R = RGB.substring(0, SP1);
    String G = RGB.substring(SP1+1, SP2);
    String B = RGB.substring(SP2+1, SP3);

    //Cetak info debug pada jendela output Serial
    Serial.print("R=");
    Serial.println( constrain(R.toInt(), 0, 255));
    Serial.print("G=");
    Serial.println(constrain(G.toInt(), 0, 255));
    Serial.print("B=");
    Serial.println( constrain(B.toInt(), 0, 255));
    //nyalakan LED dengan kode

    // pengaturan cerah redup pilihan, sett dari 0-255
    analogWrite(PIN_RED, (0+R.toInt()));
    analogWrite(PIN_GREEN, (0+G.toInt()));
    analogWrite(PIN_BLUE, (0+B.toInt()));

}

```

(c) Bagian tiga

Gambar 4.23 Pemrograman IDE untuk aplikasi

Pengujian untuk mengetahui jarak sensor *bluetooth* bekerja akan ditampilkan pada tabel 4.6.

Tabel 4.6 Pengujian Jarak Sensor *Bluetooth*

| Jarak (m) | Status | Keterangan |
|-----------|------------------|-----------------------|
| 1 | Terkoneksi | Dapat digunakan |
| 2 | Terkoneksi | Dapat digunakan |
| 3 | Terkoneksi | Dapat digunakan |
| 4 | Terkoneksi | Dapat digunakan |
| 5 | Terkoneksi | Dapat digunakan |
| 6 | Terkoneksi | Dapat digunakan |
| 7 | Terkoneksi | Dapat digunakan |
| 8 | Terkoneksi | Dapat digunakan |
| 9 | Terkoneksi | Dapat digunakan |
| 10 | Terkoneksi | Dapat digunakan |
| 11 | Tidak Terkoneksi | Tidak dapat digunakan |
| 12 | Tidak Terkoneksi | Tidak dapat digunakan |
| 13 | Tidak Terkoneksi | Tidak dapat digunakan |

Dari tabel 4.6 dapat dianalisis bahwa jarak sensor *bluetooth* untuk dapat terkoneksi dan digunakan yaitu maksimal pada jarak 10 meter. Setelah lebih dari jarak tersebut koneksi terputus dan alat tidak dapat dioperasikan untuk mengontrol pencahayaan.

4.5 Pengujian Sistem Kontrol Pencahayaan

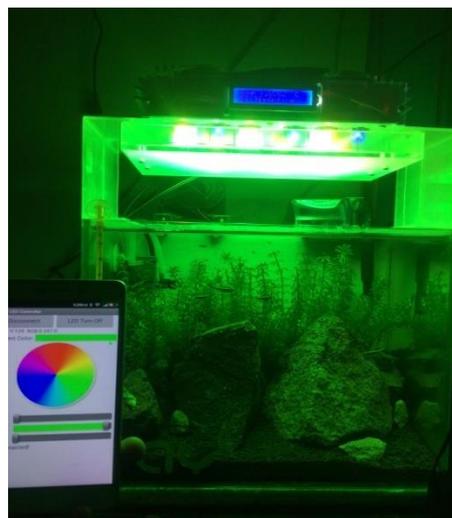
Pengujian sistem kontrol pencahayaan dilakukan untuk mengetahui apakah alat dapat bekerja dengan optimal atau tidak. Pengujian dilakukan pada *aquarium* dengan ukuran $p = 30\text{cm}$ $l = 20\text{cm}$ $t = 25\text{cm}$. Hasil pengujian alat akan ditampilkan pada gambar 4.24.



(a) Pada saat warna merah



(b) Pada saat warna biru



(c) Pada saat warna hijau



(d) Pada saat warna biru dan hijau

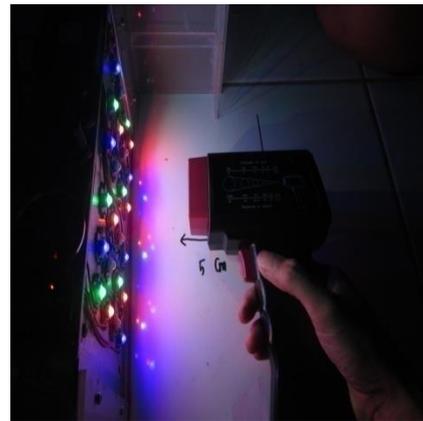
Gambar 4.24 Hasil cahaya sistem kontrol pencahayaan

Pada gambar 4.25 adalah pengambilan sampel hasil warna dari cahaya yang dihasilkan oleh HPL. Pada gambar (a) adalah hasil pada saat dikontrol untuk menghasilkan warna merah, gambar (b) untuk warna biru, gambar (c) untuk warna biru, dan gambar (d) adalah hasil perpaduan untuk warna hijau dan biru. Dari pengujian yang dilakukan dapat ditarik kesimpulan bahwa alat bekerja dengan optimal karena hasil dari cahaya HPL sesuai dengan pengontrolan warna yang diinginkan.

Dengan sistem kontrol pencahayaan ini, pengguna akan lebih mudah dalam melakukan monitoring serta pengontrolan cahaya sesuai dengan apa yang diinginkan. Selain itu, penggunaan HPL akan lebih efisien dibandingkan menggunakan jenis lampu lainnya. Pada gambar 4.25 akan ditampilkan pengujian perbedaan penggunaan HPL dengan lampu lainnya. Kemudian pada tabel 4.7 akan ditampilkan hasil pengujian perbedaan penggunaan HPL dengan lampu lainnya.



(a) Pengukuran LUX lampu HPL



(b) Pengukuran suhu lampu HPL



(c) Pengukuran LUX lampu LED BULB



(d) Pengukuran suhu lampu LED BULB

Gambar 4.25 Pengujian perbedaan HPL dengan lampu lainnya

Pada gambar 4.25 adalah pengujian yang dilakukan untuk mengetahui perbedaan HPL dengan lampu lainnya. Pengujian dilakukan di tempat gelap untuk mengukur intensitas cahaya yang dihasilkan lampu tersebut dan suhu yang

dihasilkan. Jarak untuk mengukur intensitas cahaya adalah 15cm dan untuk mengukur suhu adalah 5 cm. Hasil pengukuran akan ditampilkan pada tabel 4.7.

Tabel 4.7 Hasil Pengujian Lampu

| Merk Lampu | Tipe | Daya (W) | Suhu Ruangan (°C) | Suhu Lampu (°C) | LUX (Candela) |
|------------|----------------------------|----------|-------------------|-----------------|----------------|
| BLC | HPL (tanpa tutup) | 18 | 30 | 30,4 | 840 |
| BLC | HPL (Menggunakan tutup) | 18 | 32 | 31,1 | 840 |
| Philips | Fluorescent | 28 | 30 | 63,3 | 665 |
| Osram | Fluorescent | 18 | 30 | 41,1 | 493 |
| DOP | Pijar | 36 | 30 | 118 | 560 |
| Philips | Neon Bulb | 24 | 30 | 46,3 | 511 |

Pada tabel 4.7 telah dilakukan beberapa sampel pengujian pada jenis lampu yang berbeda. Dari hasil yang didapat dapat dianalisis bahwa pada lampu HPL menghasilkan nilai intensitas cahaya yang tinggi dibandingkan dengan lampu lainnya. Selain itu pada HPL menghasilkan suhu yang lebih rendah. Dengan penggunaan daya 18Watt, HPL memiliki nilai intensitas cahaya yang lebih tinggi dibanding lampu pijar dengan daya 36 Watt atau lampu Fluorescent dengan daya 28 Watt. Sehingga penggunaan HPL akan lebih efisien karena penggunaan listrik akan lebih irit dengan hasil pencahayaan yang optimal dan suhu kerja yang rendah. Karena jarak lampu dengan air pada *aquarium* berjarak 10 cm , maka suhu lampu akan berpengaruh terhadap suhu air pada *aquarium*. Oleh karena itu penggunaan HPL lebih optimal dibandingkan dengan penggunaan lampu jenis lainnya.