Laporan Praktikum Sensor dan Sistem Mikroprosessor

“ Arduino Uno R3 Strater Kit Latihan 1 – 17 ”

Diajukan untuk memenuhi Laporan Akhir Semester mata kuliah Praktikum Sensor dan Sistem Mikroprosessor yang diampu oleh Drs. Yoyo Somantri, S.T., M.Pd. dan Erik Haritman, S.Pd., M.T.

Disusun oleh

Ibrohim (1806068)

ELEKTRONIKA INDUSTRI 2018

PENDIDIKAN TEKNIK ELEKTRO

DEPARTEMEN PENDIDIKAN TEKNIK ELEKTRO

FAKULTAS PENDIDIKAN TEKNOLOGI DAN KEJURUAN

UNIVERSITAS PENDIDIKAN INDONESIA

BANDUNG

2021

Daftar Isi

Latihan 1 : LED Blinking 3
Latihan 2 : Trafic Light 7
Latihan 3 : Interactive Trafic Light 11
Latihan 4 : Pushbutton 16
Latihan 5 : Pushbutton Switch 20
Latihan 6 : Two Button and LED 24
Latihan 7 : Light Sensor 30
Latihan 8 : Potensiometer 36
Latihan 9 : Servo 40
Latihan 10 : Kontrol Servo 45
Latihan 11 : LCD 16x2 49
Latihan 12 : Ultrasonic 56
Latihan 13 : Ultrasonic Reverse Warning System 63
Latihan 14 : Relay 68
Latihan 15 : DHT Temperature and Humadity 72
Latihan 16 : Infrared 77
Latihan 17 : Control LED Using Infrared 81

Praktikum Sensor dan Sistem Mikroprosessor

Arduino Uno R3 Strater Kit

Latihan 1

Nama : Ibrohim
NIM : 1806068
Kelas : Elektronika Industri 2018

Judul : LED Blinking

Tujuan:

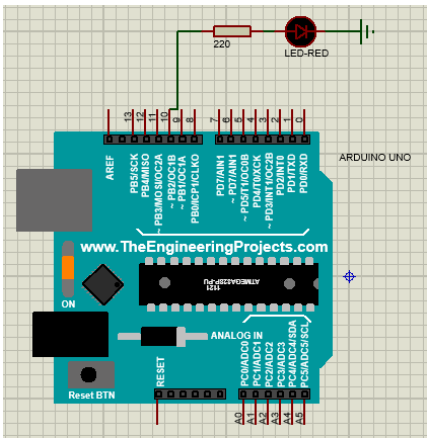
1. Menggunakan Arduino sebagai output.
2. Membuat program sederhana untuk membuat LED pada pin 10 berkedip.

Alat dan Bahan :

1. Breadboard 1 buah
2. LED 1 buah
3. Resistor 220 ohm 1 buah
4. Kabel jumper secukupnya

Langkah Kerja:

1. Siapkan alat dan bahan
2. Buatlah rangkaian seperti di bawah ini pada breadboard.



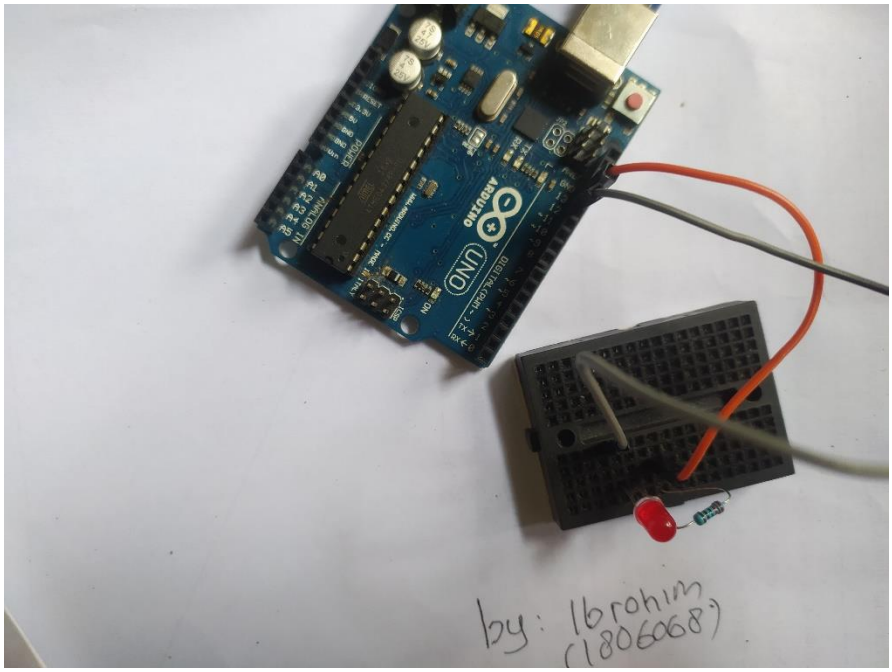
3. Masukan kode program di bawah ini ke Arduino IDE.

```
//LED Blinking
int ledPin = 10;
void setup()
{
  pinMode(ledPin, OUTPUT);
}
void loop()
{
  digitalWrite(ledPin, HIGH);
  delay(1000);
  digitalWrite(ledPin, LOW);
  delay(1000);
}
```

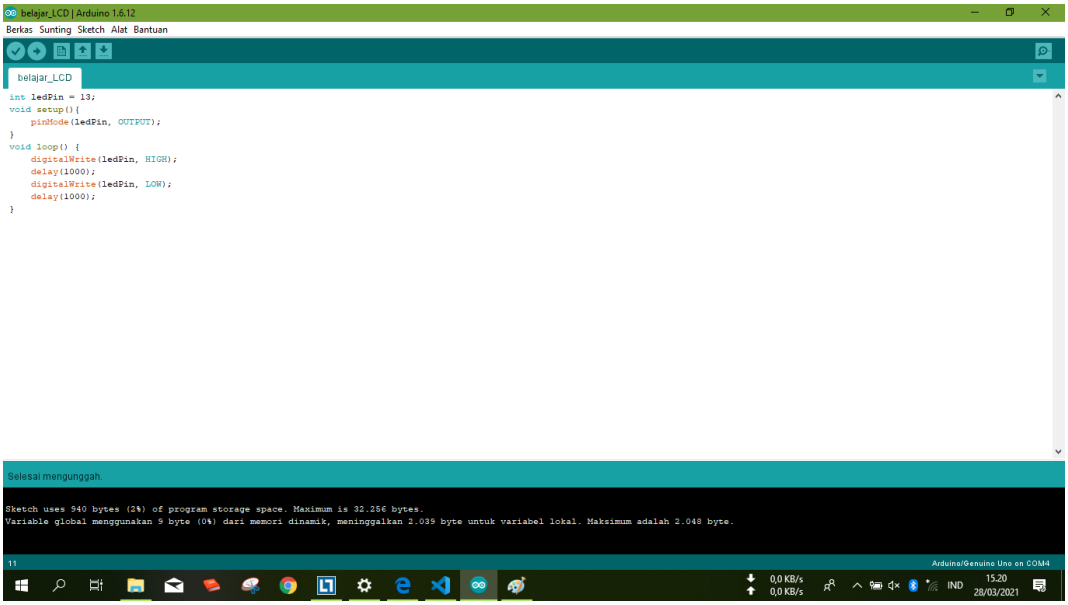
4. Compile dan upload kode program tersebut ke board Arduino Uno.
5. Amati apa yang terjadi pada praktikum ini dan buatlah kesimpulan.

Hasil Praktikum:

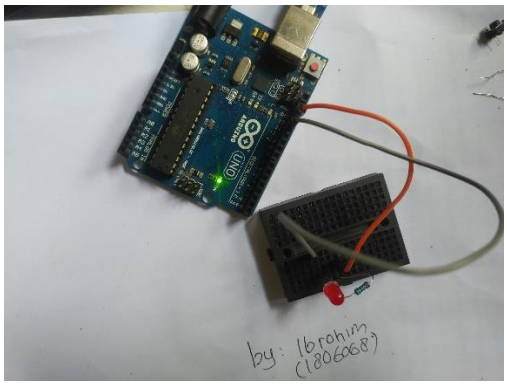
1. Hasil rangkaian pada breadboard.



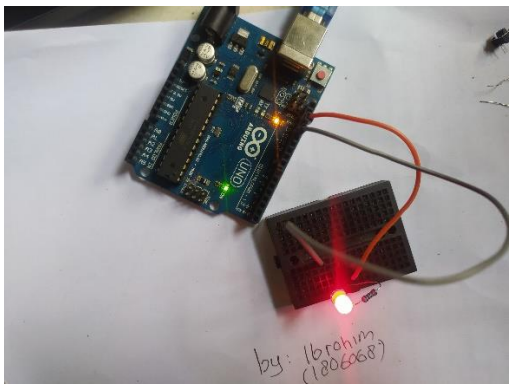
2. Hasil kode program yang sudah berhasil di-compile.



3. Hasil screenshot dokumentasi video praktikum saat kode program pada Arduino IDE berhasil di-upload ke board Arduino.



LED mati selama 1000 ms (1 sekon)



LED hidup selama 1000 ms (1 sekon)

Analisis

Pada rangkaian sederhana ini terdapat kode program yang terbagi menjadi 3 blok, yaitu Header, Setup, dan Loop. Pada bagian header ditulis pendefinisian variable, yakni variable led (integer) yang diisi dengan angka 10 (pin I/O 10). Pada bagian setup ditulis sebuah pin sebagai input atau ouput dengan menggunakan perintah pinMode. Terakhir, pada bagian loop ditulis kode program yang akan dieksekusi terus menerus. Apabila program sudah sampai akhir blok maka akan terus mengulang dari awal blok. Perintah digitalWrite(pinNumber,nilai)akan memerintahkan Arduino untuk menyalakan atau mematikan tegangan di pinNumber tergantung nilainya. Jadi perintah digitalWrite(led,HIGH) akan membuat pin nomor 10 memiliki tegangan = 5V (HIGH). Hanya ada dua kemungkinan nilai digitalWrite yaitu HIGH atau LOW yang sebetulnya adalah nilai integer 1 atau 0.

Kesimpulan:

Berasarkan praktikum LED Blinking ini, kita bisa membuat output LED nyala atau mati dengan menggunakan perintah digitalWrite pada void loop yang tentu sebelumya telah inisialisasi terlebih dahulu. Dan menambahkan fungsi delay(1000) yang sebelumya led di set

high lalu dilanjutkan dengan led di set low maka kita dapat membuat program LED Blinking. Dan progam ini umumnya dipakai untuk menguji arduino/lainya saat pertama kali digunakan (test alat).

Praktikum Sensor dan Sistem Mikroprosessor

Arduino Uno R3 Strater Kit

Latihan 2

Nama : Ibrohim
NIM : 1806068
Kelas : Elektronika Industri 2018

Judul : Trafic Light

Tujuan:

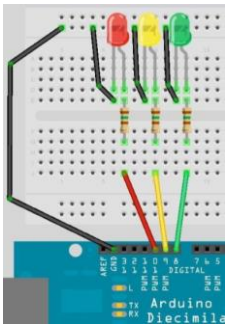
1. Menggunakan Arduino sebagai output (3 output).
2. Membuat program sederhana untuk membuat LED pada pin 10, 9, dan 8 berkedip/bernilai high lalu low sehingga bisa seperti Trafic Light.
3. Menggunakan fungsi delay.

Alat dan Bahan :

1. Breadboard 1 buah
2. LED Merah 1 buah
3. LED Kuning 1 buah
4. LED Hijau 1 buah
5. Resistor 220 ohm 1 buah
6. Kabel jumper secukupnya

Langkah Kerja:

1. Siapkan alat dan bahan
2. Buatlah rangkaian seperti di bawah ini pada breadboard.



Kali ini kita menghubungkan 3 LED sekaligus dengan pin Anoda LED terhubung dengan pin 8, 9 dan 10 papan arduino melalui resistor 220 Ω sebelumnya. Untuk memudahkan merangkai maka kita hubungkan semua pin katoda led pada satu jalur breadboard menuju pin ground papan arduino

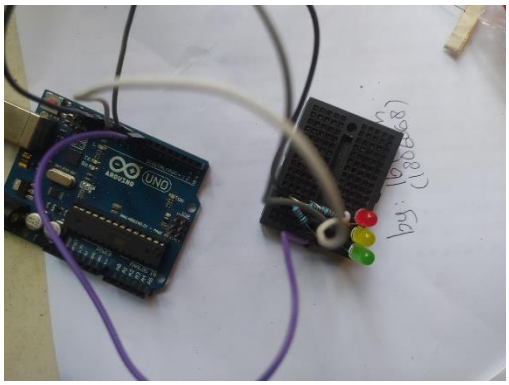
3. Masukan kode program di bawah ini ke Arduino IDE.

```
//Traffic Lights
int ledDelay = 5000; // delay in between changes
int redPin = 10;
int yellowPin =9;
int greenPin =8;

void setup()
{
  pinMode(redPin,OUTPUT);
  pinMode(yellowPin,OUTPUT);
  pinMode(greenPin,OUTPUT);
}
void loop() {
// turn the red light on
  digitalWrite(redPin, HIGH);
  delay(ledDelay); // wait 5 seconds
  digitalWrite(yellowPin, HIGH); // turn on yellow
  delay(2000); // wait 2 seconds
  digitalWrite(greenPin, HIGH); // turn green on
  digitalWrite(redPin, LOW); // turn red off
  digitalWrite(yellowPin, LOW); // turn yellow off
  delay(ledDelay); // wait ledDelay milliseconds
  digitalWrite(yellowPin, HIGH); // turn yellow on
  digitalWrite(greenPin, LOW); // turn green off
  delay(2000); // wait 2 seconds
  digitalWrite(yellowPin, LOW); // turn yellow off
// now our loop repeats
}
```

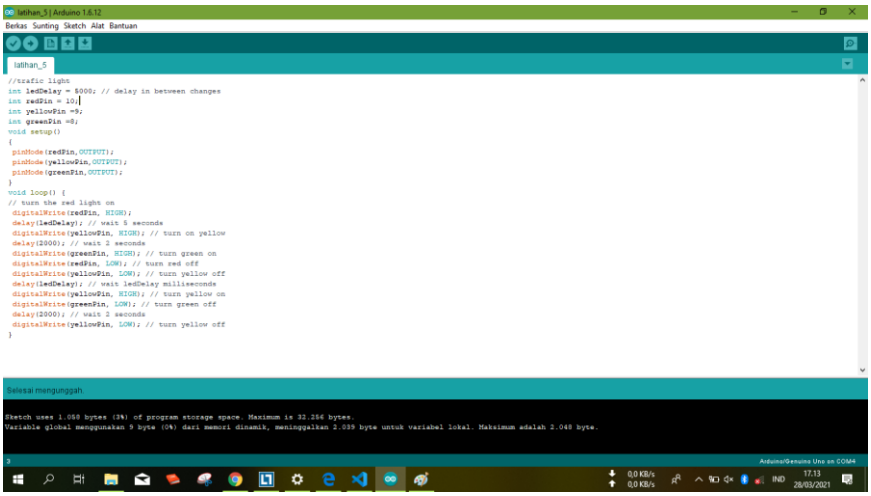
4. Compile dan upload kode program tersebut ke board Arduino Uno.
5. Amati apa yang terjadi pada praktikum ini dan buatlah kesimpulan.

Hasil Praktikum:

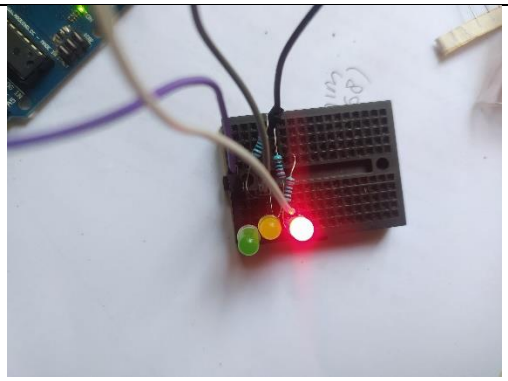
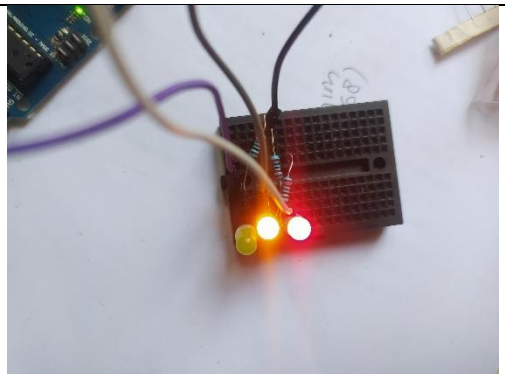
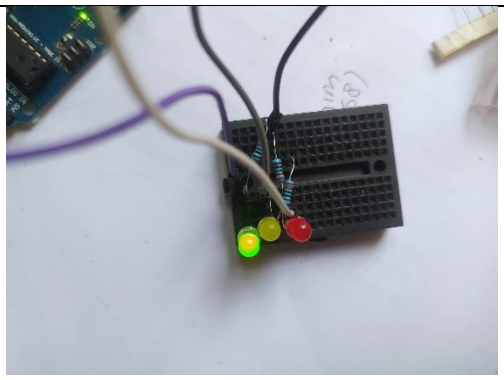
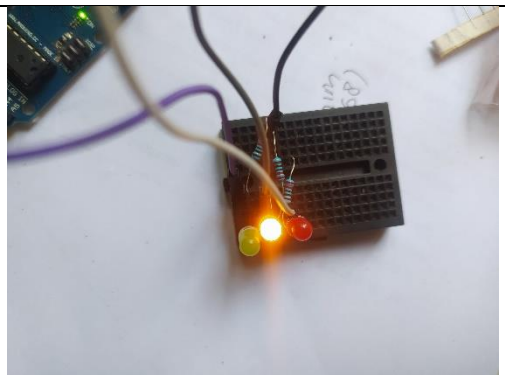
1. Hasil rangkaian pada breadboard.



2. Hasil kode program yang sudah berhasil di-compile.



3. Hasil screenshot dokumentasi video praktikum saat kode program pada Arduino IDE berhasil di-upload ke board Arduino.

 LED merah nyala selama 5000 ms (5 sekon) melalui fungsi delay(ledDelay)	 Led merah dan kuning nyala selama 2 detik
 Led merah dan kuning mati, dan Led hijau menyala selama 5 detik	 Led hijau mati, lalu Led kuning menyala selama 2 detik

Analisis

Pada rangkaian sederhana ini terdapat kode program yang terbagi menjadi 3 blok, yaitu Header, Setup, dan Loop. Pada bagian header ditulis pendefinisian variable, yakni variable ledDelay yang diisi dengan 5 sekon (5000ms), variabel redPin (integer) sebagai Led Merah yang diisi dengan pin 10 (pin I/O 10), variabel yellowPin (integer) sebagai Led Kuning yang diisi dengan pin 9 (pin I/O 9), variabel greenPin (integer) sebagai Led Hijau yang diisi dengan pin 8 (pin I/O 8).

Pada bagian setup ditulis sebuah pin sebagai ouput dengan menggunakan perintah pinMode yakni pin 10, 9, dan 8 (pin semua LED). Terakhir, pada bagian loop ditulis kode program yang akan dieksekusi terus menerus. Apabila program sudah sampai akhir blok maka akan terus mengulang dari awal blok. Perintah digitalWrite(pinNumber,nilai)akan memerintahkan Arduino untuk menyalakan atau mematikan tegangan di pinNumber tergantung nilainya. Jadi perintah digitalWrite(led,HIGH) akan membuat pin nomor 10 atau 9 atau 8 memiliki tegangan = 5V (HIGH). Hanya ada dua kemungkinan nilai digitalWrite yaitu HIGH atau LOW yang sebetulnya adalah nilai integer 1 atau 0. Pertama, Led merah di set bernilai high (digitalWrite(redPin, HIGH)) Lalu setelah bernilai high, fungsi delay(ledDelay) dipanggil sehingga akan menunda 5 detik (karena nilainya 5000), setelah itu led kuning di set high (digitalWrite(yellowPin, HIGH)) lalu memanggil fungsi delay(200) atau 2 detik setelah itu, Led hijau di set high (digitalWrite(greenPin, HIGH)) bersamaan dengan led merah dan

kuning di set low sehingga dapat disimpulkan led merah menyala selama 7 detik dan led kuning 2 detik. Lalu setelah itu dipanggil kembali ledDelay bersamaan dengan led kuning di set high dan led hijau di set low sehingga led hijau hanya menyala selama 5 detik lalu menyalakan led kuning. Lalu memanggil delay(2000) dan mengeset led kuning ke low sehingga led kuning hanya menyala selama 2 detik. Setekah itu program akan di ulang dari awal lagi (void loop).

Kesimpulan:

Berdasarkan praktikum Trafic Light ini, kita dapat mengetahui penggunaan 3 output arduino dan fungsi delay sehingga dapat membuat program Trafic Light. Dimana pada program ini, akan menyalakan Led merah selama 7 detik bersamaan dengan Led kuning 2 detik, setelah itu mematikan kedua led tersebut dan menyalakan led hijau selama 5 detik, dan mematikan led hijau bersamaan dengan menyallakan led kuning selama 2 detik, serta mengulanginya lagi dari awal.

Praktikum Sensor dan Sistem Mikroprosessor

Arduino Uno R3 Strater Kit

Latihan 3

Nama : Ibrohim
NIM : 1806068
Kelas : Elektronika Industri 2018

Judul : Interactive Trafic Light

Tujuan :

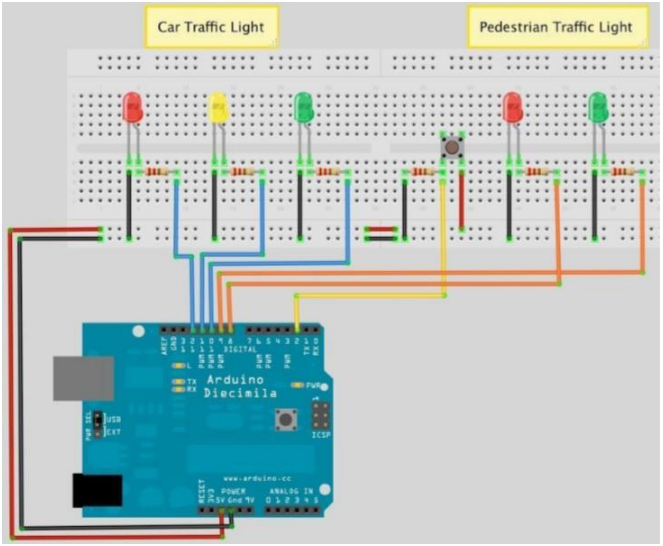
1. Menggunakan Arduino sebagai output (5 output).
2. Mengatur Output dengan Input Push Button.
3. Membuat program untuk mengatur Trafic Light antara Mobil dan Pejalan Kaki.
4. Menggunakan fungsi delay dan milis.

Alat dan Bahan :

1. Breadboard 1 buah
2. LED Merah 2 buah
3. LED Kuning 1 buah
4. LED Hijau 2 buah
5. Resistor 220 ohm 5 buah
6. Kabel jumper secukupnya
7. Push button 1 buah
8. Resistor 10k ohm 1 buah

Langkah Kerja:

1. Siapkan alat dan bahan
2. Buatlah rangkaian seperti di bawah ini pada breadboard.



Hubungkan LED dan push button sesuai dengan diagram yang ada pada halaman sebelumnya, lalu hubungkan LED pada pin 8,9,10,11 dan 12 serta push button pada pin 2 yang di pullup menggunakan resistor 10k

3. Masukan kode program di bawah ini ke Arduino IDE.

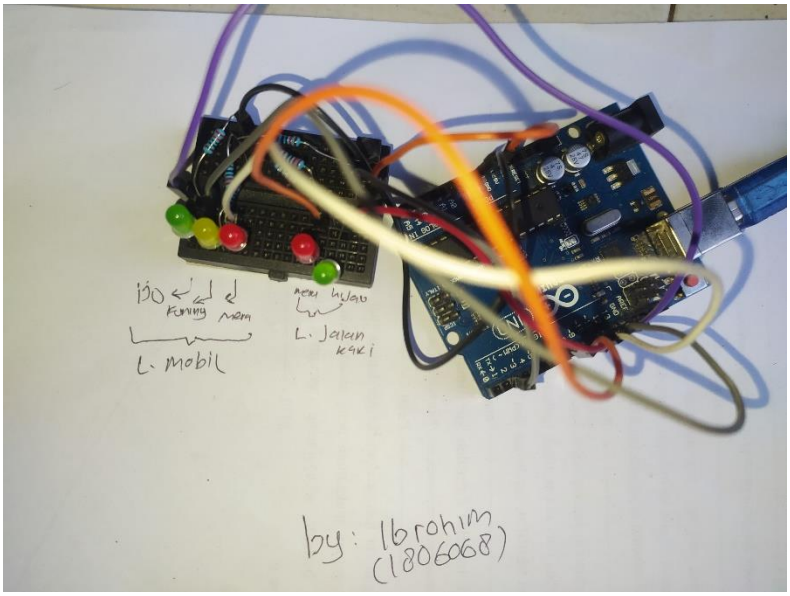
```
//Interactive Traffic Lights
int carRed = 12; // assign the car lights
int carYellow = 11;
int carGreen = 10;
int pedRed = 9; // assign the pedestrian lights
int pedGreen = 8;
int button = 2; // button pin
int crossTime = 5000; // time allowed to cross
unsigned long changeTime; // time since button pressed

void setup()
{
  pinMode(carRed, OUTPUT);
  pinMode(carYellow,OUTPUT);
  pinMode(carGreen, OUTPUT);
  pinMode(pedRed, OUTPUT);
  pinMode(pedGreen, OUTPUT);
  pinMode(button, INPUT); // button on pin 2
// turn on the green light
  digitalWrite(carGreen, HIGH);
  digitalWrite(pedRed, HIGH);
}

void loop() {
int state = digitalRead(button);
/* check if button is pressed and it is over 5 seconds
since last button press */
  if (state==HIGH && (millis()- changeTime) >5000)
  {
// Call the function to change the lights
    changeLights();
  }
}
```

```
void changeLights()
{
  digitalWrite(carGreen, LOW); // green off
  digitalWrite(carYellow, HIGH); // yellow on
  delay(2000); // wait 2 seconds

  digitalWrite(carYellow, LOW); // yellow off
  digitalWrite(carRed, HIGH); // red on
  delay(1000); // wait 1 second till its safe
  digitalWrite(pedRed, LOW); // ped red off
  digitalWrite(pedGreen, HIGH); // ped green on
  delay(crossTime); // wait for preset time period
// flash the ped green
  for (int x=0; x<10; x++)
  {
    digitalWrite(pedGreen, HIGH);
    delay(250);
    digitalWrite(pedGreen, LOW);
    delay(250);
  }
// turn ped red on
  digitalWrite(pedRed, HIGH);
  delay(500);
  digitalWrite(carYellow, HIGH); // yellow on
  digitalWrite(carRed, LOW); // red off
  delay(1000);
  digitalWrite(carGreen, HIGH);
  digitalWrite(carYellow, LOW); // yellow off
// record the time since last change of lights
  changeTime = millis();
// then return to the main program loop
}
```

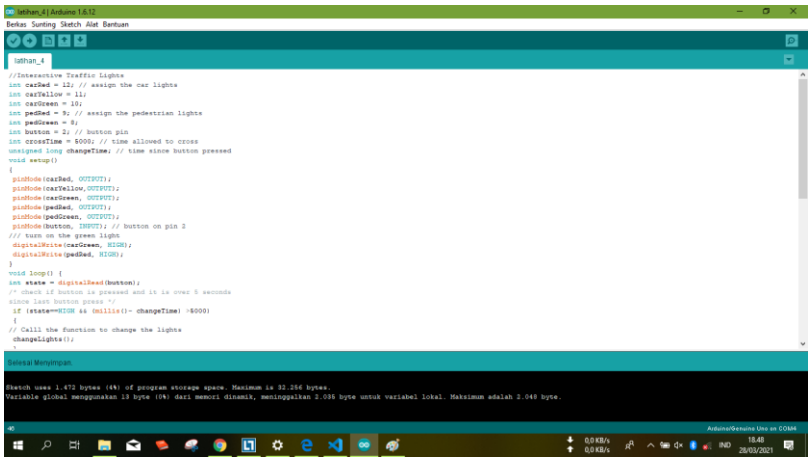
4. Compile dan upload kode program tersebut ke board Arduino Uno.
5. Amati apa yang terjadi pada praktikum ini dan buatlah kesimpulan.

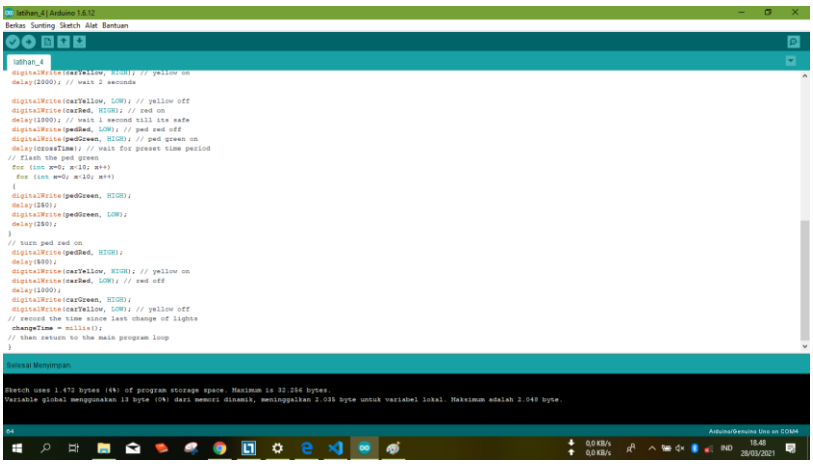
Hasil Praktikum:

1. Hasil rangkaian pada breadboard.

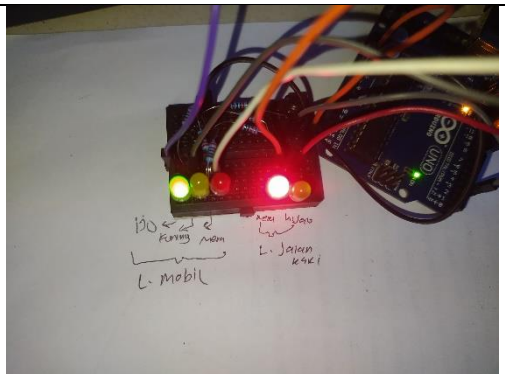


2. Hasil kode program yang sudah berhasil di-compile.

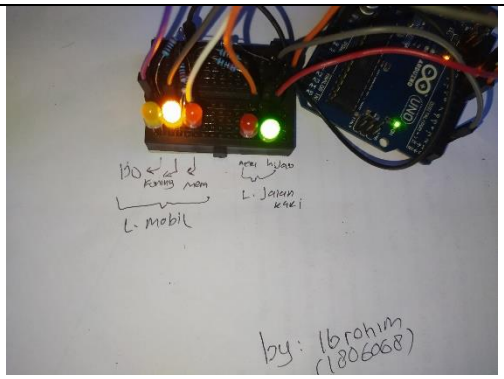


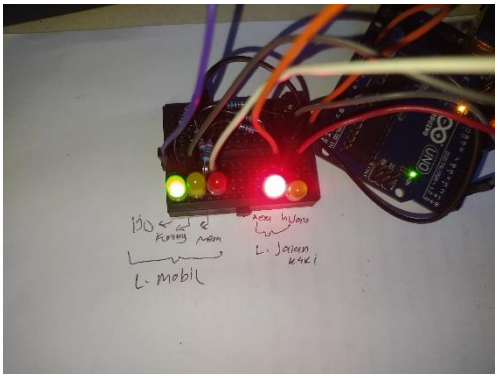


3. Hasil screenshot dokumentasi video praktikum saat kode program pada Arduino IDE berhasil di-upload ke board Arduino.

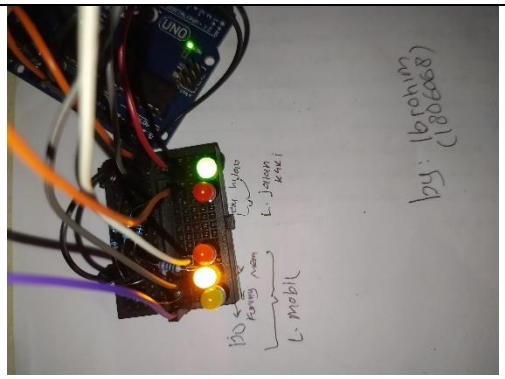


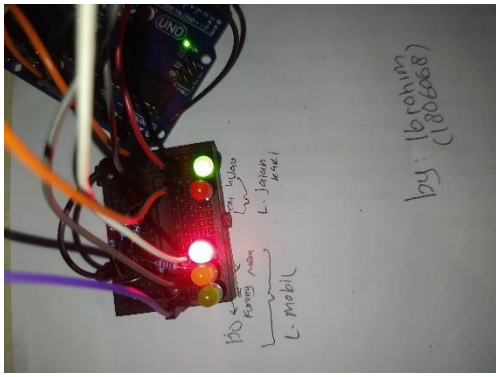
Kondisi awal (Sebelum tombol ditekan) LED Hijau mobil menyala dan LED merah pejalan kaki menyala (Lampu merah pejalan kaki dan lampu hijau mobil)





Setelah beberapa saat LED kuning mobil menyala dan LED hjau Pejalan kaki tetap hidup (2 sekon, menandakan kondisi bersiap berganti lampu hijau mobil) lalu kemudian berganti LED lampu merah pejalan kaki menyala (dengan nyala kedip) dan led lampu hijau mobil menyala (Kondisi lampu merah untuk pejalan kaki dan lampu hijau untuk mobil)





Setelah Push Button di tekan, LED kuning mobil menyala (2 sekon, menandakan kondisi bersiap berganti lampu merah mobil) lalu kemudian berganti LED lampu merah mobil menyala dan led lampu hijau pejalan kaki menyala (Kondisi lampu merah untuk mobil dan lampu hijau untuk pejalan kaki)

Analisis

Pada rangkaian ini terdapat kode program yang terbagi menjadi 4 blok, yaitu Header, Setup, dan Loop, serta changeLights. Pada bagian header ditulis pendefinisian variable, Pada bagian setup ditulis sebuah pin sebagai input atau ouput, pada bagian loop ditulis kode program yang akan dieksekusi seperti apa dengan kondisi tertentu yang terus menerus, dan changeLigths yang berfungsi untuk mengatur nyala kondisi LED untuk Lampu merah Mobil dan LED untuk Lampu Merah pejalan kai.

Dimana prinsipnya, ketika menjalakan program kita akan melihat trafic light untuk mobil akan berwarna hijau, dan trafic light untuk pejalan kaki berwarna merah. Ketika push button kita tekan, program akan mengecek keadaan trafic light untuk mobil agar berubah menjadi merah, setelah itu giliran lampu untuk pejalan kaki yang akan berubah warna menjadi hijau, waktu tunggu yang di tentukan dari mulai penekanan tombol hingga perubahan trafic light menjadi warna merah di atur oleh syntax changeLight() sedangkan untuk lamanya waktu yang dibutuhkan untuk lampu pejalan kaki dari berwarna merah menjadi hijau diatur oleh syntax changeTime() pada source code di halaman 2. Lampu pejalan kaki akan berwarna hijau lalu berkedip maka waktu tunda untuk lampu pejalan kaki akan habis, dan lampu akan berwarna merah dan trafic light untuk mobil akan berubah menjadi warna kuning dan hijau kembali.

Kesimpulan:

Berdasarkan praktikum Trafic Light Interactive ini, kita dapat mengetahui penggunaan 5 output arduino yang dihubungkan dengan input push button dan fungsi delay serta fungsi milis() sehingga dapat membuat program Trafic Light yang interaktif antara mobil dan pejalan kaki. Dimana pada program ini, ketika menjalakan program kita akan melihat trafic light untuk mobil akan berwarna hijau, dan trafic light untuk pejalan kaki berwarna merah. Ketika push button kita tekan, program akan mengecek keadaan trafic light untuk mobil agar berubah menjadi merah, setelah itu giliran lampu untuk pejalan kaki yang akan berubah warna menjadi hijau, waktu tunggu yang di tentukan dari mulai penekanan tombol hingga perubahan trafic light menjadi warna merah di atur oleh syntax changeLight() sedangkan untuk lamanya waktu yang dibutuhkan untuk lampu pejalan kaki dari berwarna merah menjadi hijau diatur oleh syntax changeTime() pada source code di halaman 2. Lampu pejalan kaki akan berwarna hijau lalu berkedip maka waktu tunda untuk lampu pejalan kaki akan habis, dan lampu akan berwarna merah dan trafic light untuk mobil akan berubah menjadi warna kuning dan hijau kembali.

Praktikum Sensor dan Sistem Mikroprosessor

Arduino Uno R3 Strater Kit

Latihan 4

Nama : Ibrohim
NIM : 1806068
Kelas : Elektronika Industri 2018

Judul : Pushbutton

Tujuan :

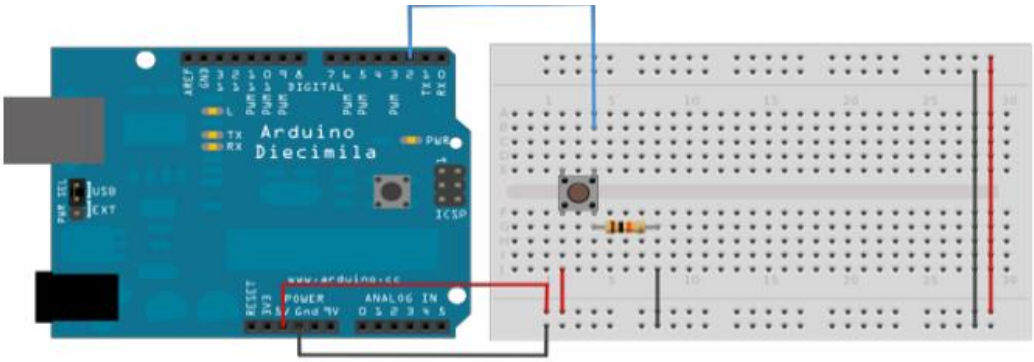
1. Menggunakan Arduino sebagai input.
2. Membuat program sederhana dengan push button sebagai input untuk menyalakan dan mematikan nyala lampu LED.

Alat dan Bahan :

1. Breadboard 1 buah
2. LED Merah 1 buah
3. Resistor 220 ohm 1 buah
4. Kabel jumper secukupnya
5. Push button 1 buah
6. Resistor 10k ohm 1 buah

Langkah Kerja:

1. Siapkan alat dan bahan
2. Buatlah rangkaian seperti di bawah ini pada breadboard.



Pertama hubungkan 1 kaki pushbutton dihubungkan dengan 1 kaki resistor dan pin digital 2, kaki pushbutton yang lain dihubungkan dengan VCC dan satu kaki resistor yang lain di hubungkan dengan ground. Lihat contoh diagram rangkaian di atas. Sedangkan untuk led kita bisa menghubungkannya dengan ke kaki 13 setelah melalui resistor 220Ω, sedangkan kaki katoda LED dimasukan ke ground arduino, pada

project ini bisa digunakan juga LED internal yang telah disediakan di papan arduino, sehingga kita tidak perlu menambahkan led lagi.

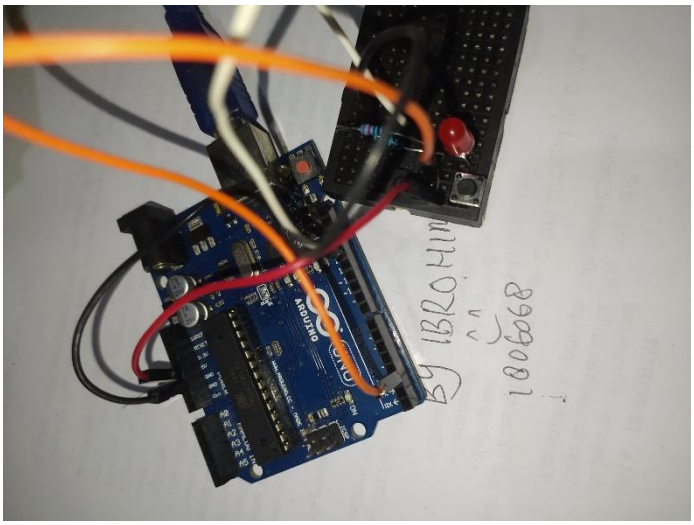
3. Masukan kode program di bawah ini ke Arduino IDE.

```
//Pushbutton

const int buttonPin = 2; //inisialisasi pin button
const int ledPin = 13; //inisialisasi pin LED
int buttonState = 0; //variable
// variable digunakan untuk pembacaan status button

void setup()
{
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
}

void loop() {
//membaca nilai dari pin button
  buttonState = digitalRead(buttonPin);

//mengecek jika pushbutton ditekan
//buttonState benilai HIGH atau di tekan maka,
  if (buttonState == HIGH)
  {
    //LED Menyala
    digitalWrite(ledPin, HIGH);}
  else {
    // turn LED off:
  digitalWrite(ledPin, LOW);
  }
}
```

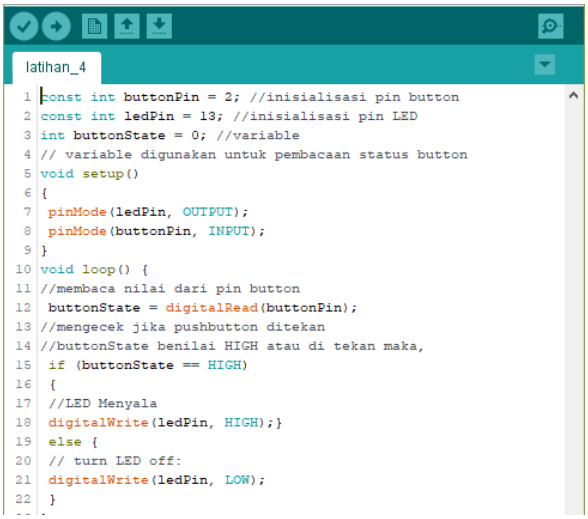
4. Compile dan upload kode program tersebut ke board Arduino Uno.
5. Amati apa yang terjadi pada praktikum ini dan buatlah kesimpulan.

Hasil Praktikum:

1. Hasil rangkaian pada breadboard.



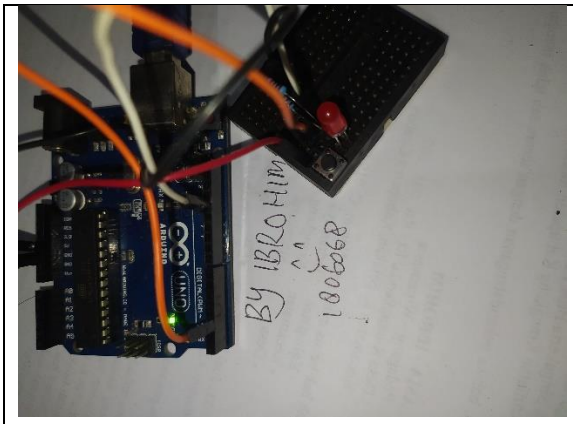
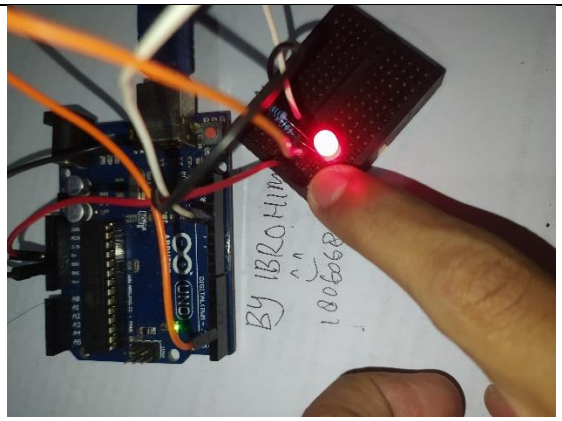
2. Hasil kode program yang sudah berhasil di-compile.



latihan_4

```
const int buttonPin = 2; //inisialisasi pin button
const int ledPin = 13; //inisialisasi pin LED
int buttonState = 0; //variable
// variable digunakan untuk pembacaan status button
void setup()
{
 pinMode(ledPin, OUTPUT);
 pinMode(buttonPin, INPUT);
}
void loop() {
//membaca nilai dari pin button
 buttonState = digitalRead(buttonPin);
//mengecek jika pushbutton ditekan
//buttonState benilai HIGH atau di tekan maka,
 if (buttonState == HIGH)
 {
 //LED Menyala
 digitalWrite(ledPin, HIGH);}
 else {
 // turn LED off:
 digitalWrite(ledPin, LOW);
 }
```

3. Hasil screenshot dokumentasi video praktikum saat kode program pada Arduino IDE berhasil di-upload ke board Arduino.

	
Kondisi awal saat pushbutton tidak ditekan atau dilepas	Kondisi saat pusbutton ditekan (ditekan terus tanpa di lepas)

Analisis

Inisialisasi variable, dalam hal ini inisialisasi variable bernama buttonPin dengan type data konstanta integer dan nilai 2 (pada pin 2) untuk pin tombolnya, variable bernama ledPin dengan type data konstanta integer dan nilai 13 untuk pin lednya, dan variable bernama buttonstate dengan type data integer dan nilai 0 (false) untuk nilai pembacaan status button/tombolnya.

```
//Pushbutton

const int buttonPin = 2; //inisialisasi pin button
const int ledPin = 13; //inisialisasi pin LED
int buttonState = 0; //variable
// variable digunakan untuk pembacaan status button
```

Pada bagian selanjutnya yakni Void setup ditulis sebuah pin sebagai input atau ouput, dengan menggunakan perintah pinmode, pada variabel ledPin di set sebagai output dan pada bagian buttonPin sebagai input.

```
void setup()
{
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
}
```

Terakhir, pada bagian loop ditulis kode program yang akan dieksekusi terus menerus. Yang mana perintah pertama adalah membaca nilai buttonPin dengan perintah digitalRead yang kemudian nilainya dimasukan ke variabel buttonState . Selanjutnya masuk ke pengkodisian if dimana jika nilai dari buttonState adalah high maka ledPin akan diubah menjadi high dengan perintah digitalWrite begitupun sebaliknya.

```
void loop() {
//membaca nilai dari pin button
  buttonState = digitalRead(buttonPin);

//mengecek jika pushbutton ditekan
//buttonState benilai HIGH atau di tekan maka,
  if (buttonState == HIGH)
  {
    //LED Menyala
    digitalWrite(ledPin, HIGH);}
  else {
    // turn LED off:
  digitalWrite(ledPin, LOW);
  }
}
```

Kesimpulan:

Berdasarkan praktikum Pushbutton ini, kita dapat mengetahui penggunaan input pushbutton pada arduino dan penggunaan output led sehingga dapat Membuat program sederhana dengan push button sebagai input untuk menyalakan dan mematikan nyala lampu LED. Juga kita telah memahami perintah untuk mengitegrasikan pushbuuton dengan led, sehingga bisa kita kembangkan untuk project lebih lanjut mengenai pushbuuton ini.

Praktikum Sensor dan Sistem Mikroprosessor

Arduino Uno R3 Strater Kit

Latihan 5

Nama : Ibrohim
NIM : 1806068
Kelas : Elektronika Industri 2018

Judul : Pushbutton Switch

Tujuan :

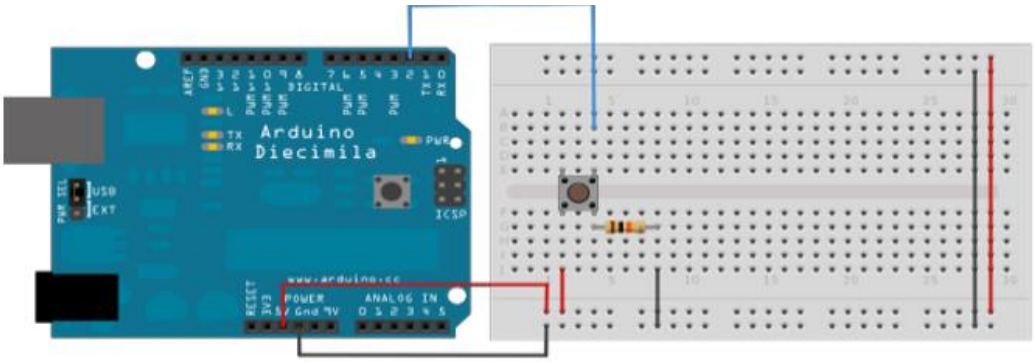
1. Menggunakan Arduino sebagai input.
2. Membuat program sederhana dengan push button sebagai input saklar/switch untuk menyalakan dan mematikan nyala lampu LED layaknya switch.

Alat dan Bahan :

1. Breadboard 1 buah
2. LED Merah 1 buah
3. Resistor 220 ohm 1 buah
4. Kabel jumper secukupnya
5. Push button 1 buah
6. Resistor 10k ohm 1 buah

Langkah Kerja:

1. Siapkan alat dan bahan
2. Buatlah rangkaian seperti di bawah ini pada breadboard.



Pertama hubungkan 1 kaki pushbutton dihubungkan dengan 1 kaki resistor dan pin digital 2, kaki pushbutton yang lain dihubungkan dengan VCC dan satu kaki resistor yang lain di hubungkan dengan ground. Lihat contoh diagram rangkaian Latihan 4. Sedangkan untuk led kita bisa menghubungkannya dengan ke kaki 13 setelah melalui resistor 220Ω, sedangkan kaki katoda LED dimasukan ke ground arduino, pada

project ini bisa digunakan juga LED internal yang telah disediakan di papan arduino, sehingga kita tidak perlu menambahkan led lagi.Masukan kode program di bawah ini ke Arduino IDE.

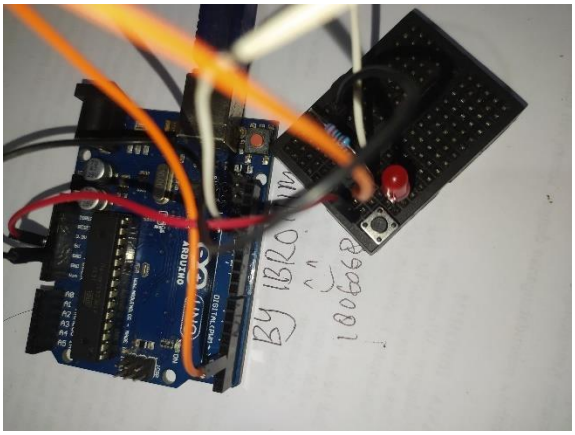
3. Compile dan upload kode program tersebut ke board Arduino Uno.

```
//Pushbutton Switch

int inPin = 2;
int outPin = 13;
int state = HIGH;
int reading;
int previous = LOW;
long time = 0;
long debounce = 200;

void setup()
{
  pinMode(inPin, INPUT);
  pinMode(outPin, OUTPUT);
}

void loop(){
  reading = digitalRead(inPin);
  if (reading == HIGH && previous == LOW && millis() - time
> debounce){
    if (state == HIGH)
      state = LOW;
    else
      state = HIGH;
    time = millis();
  }
gitalWrite(outPin, state);
previous = reading;
```

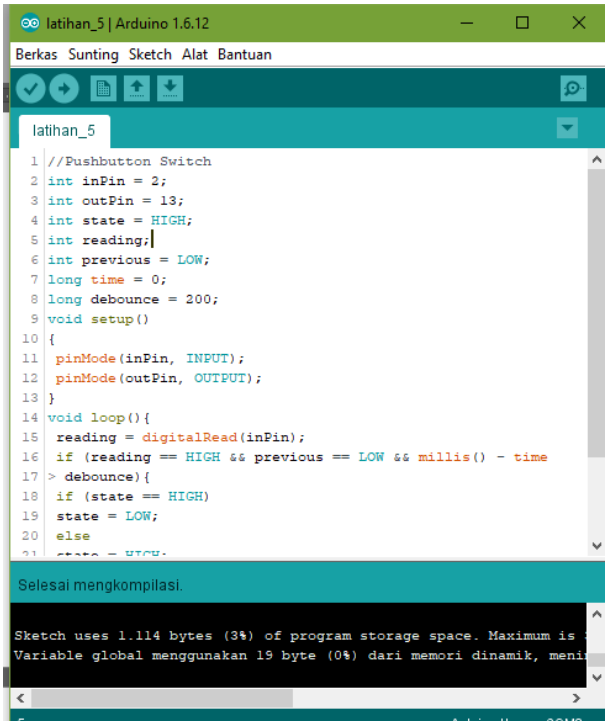
4. Amati apa yang terjadi pada praktikum ini dan buatlah kesimpulan.

Hasil Praktikum:

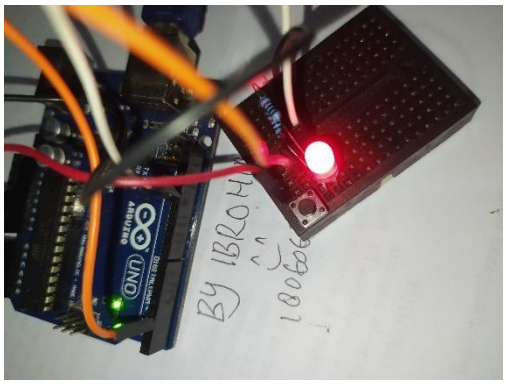
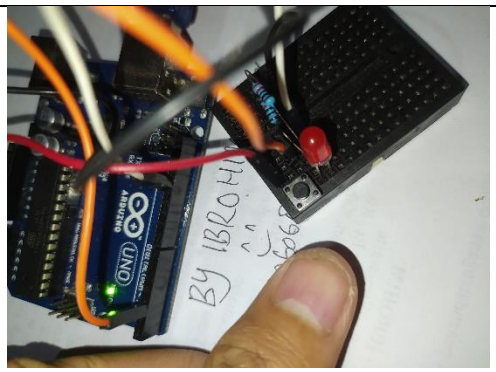
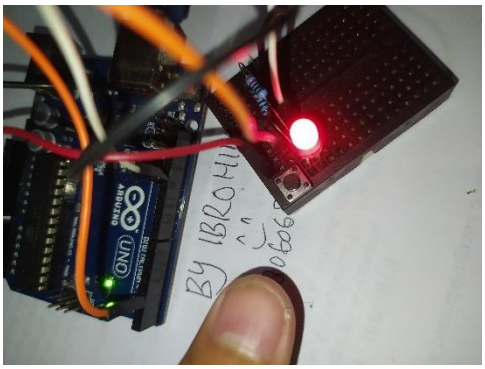
1. Hasil rangkaian pada breadboard.



2. Hasil kode program yang sudah berhasil di-compile.



3. Hasil screenshot dokumentasi video praktikum saat kode program pada Arduino IDE berhasil di-upload ke board Arduino.

 Kondisi awal saat pushbutton tidak ditekan dalam keadaan menyala.	  Kondisi saat pusbutton ditekan-lepas (switch) sekali (gambar atas) led menjadi mati . dan ditekan-lepas lagi led menyala kembali (gambar bawah)

Analisis

Inisialisasi variable, dalam hal ini inisialisasi variable bernama inPin dengan type data integer dan nilai 2 (pada pin 2) untuk pin tombolnya, variable bernama outPin dengan type data integer dan nilai 13 untuk pin lednya, variable bernama state dengan type data integer dan nilai high untuk nilai pembacaan status button/tombolnya, variable bernama reading dengan type data integer untuk pembacaan nilai button, variable bernama previous dengan type data integer dan nilai low untuk menyimpan status button sebelumnya, variable bernama time dengan type data long dan nilai awal 0 untuk pewaktu, variable bernama debounce dengan type data long dan nilai awal 200 untuk delay.

```
int inPin = 2;
int outPin = 13;
int state = HIGH;
int reading;
int previous = LOW;
long time = 0;
long debounce = 200;
```

Pada bagian selanjutnya yakni Void setup ditulis sebuah pin sebagai input atau ouput, dengan menggunakan perintah pinmode, pada variabel inPin di set sebagai input dan pada bagian outPin sebagai output.

```
void setup()
{
  pinMode(inPin, INPUT);
  pinMode(outPin, OUTPUT);
}
```

Terakhir, pada bagian loop ditulis kode program yang akan dieksekusi terus menerus. Yang mana perintah pertama adalah membaca nilai inPin dengan perintah digitalRead yang kemudian nilainya dimasukan ke variabel reading. Selanjutnya masuk ke pengkodisian if dimana jika nilai dari (reading == HIGH && previous == LOW && millis() - time > debounce) maka akan membaca kondisi if variabel state dimana jika nilainya high maka state diubah bernilai low yang akan membuat outPin bernilai low dengan perintah digitalWrite sehingga led mati dan sebaliknya. Lalu terakhir mengubah variabel previous bernilai varibel reading.

```
void loop(){
  reading = digitalRead(inPin);
  if (reading == HIGH && previous == LOW && millis() - time
> debounce){
    if (state == HIGH)
      state = LOW;
    else
      state = HIGH;
    time = millis();
  }
digitalWrite(outPin, state);
previous = reading;
}
```

Kesimpulan:

Berdasarkan praktikum Pushbutton ini, kita dapat mengetahui penggunaan input pushbutton pada arduino dan penggunaan output led sehingga dapat Membuat program sederhana dengan push button switch sebagai input untuk menyalakan dan mematikan nyala lampu LED. Juga kita telah memahami perintah untuk mengitegrasikan pushbuuton menjadi switch dengan led, sehingga bisa kita kembangkan untuk project lebih lanjut mengenai pushbuuton switch ini.

Praktikum Sensor dan Sistem Mikroprosessor

Arduino Uno R3 Strater Kit

Latihan 6

Nama : Ibrohim
NIM : 1806068
Kelas : Elektronika Industri 2018

Judul : Two Button and LED

Tujuan :

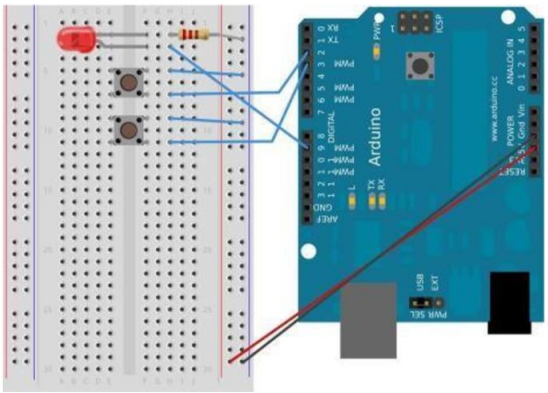
1. Menggunakan Arduino sebagai input.
2. Dengan mengubah nilai resistor yang terhubung sebagai pembatas arus untuk LED.
3. Menggunakan fungsi yang disebut dengan PWM (Pulse Width Modulation).
4. Membuat program sederhana dengan dua pushbutton sebagai input untuk mengatur kecerahan lampu LED.

Alat dan Bahan :

1. Breadboard 1 buah
2. LED Merah 1 buah
3. Resistor 220 ohm 1 buah
4. Kabel jumper secukupnya
5. Push button 2 buah
6. Resistor 10k ohm 1 buah

Langkah Kerja:

1. Siapkan alat dan bahan
2. Buatlah rangkaian seperti di bawah ini pada breadboard.



Rangkailah pushbutton, LED dan Resistor seperti pada rangkaian diatas, pushbutton 1 ke pin digital 2, pushbutton 2 ke pin digital 2 dan kaki Anoda LED ke pin PWM 9. Untuk melakukan pembuktian setelah di upload buka serial monitor arduino

IDE, setting baud rate ke 9600 (atau bisa juga memakai LCD), maka akan terlihat nilai awal 128, nilai berikut merupakan nilai awal LEDyang sudah di definisikan di progam int *ledBrightness* = 128, lalu tekan salah satu button lihat hasilnya di serial monitor arduino.

3. Compile dan upload kode program tersebut ke board Arduino Uno.

```
//Two Button and LED
const int kPinButton1 = 2;
const int kPinButton2 = 3;
const int kPinLed = 9;
int ledBrightness = 128; //nilai pertama LED

void setup()
{
  Serial.begin(9600);
  pinMode(kPinButton1, INPUT);
  pinMode(kPinButton2, INPUT);
  pinMode(kPinLed, OUTPUT);
  digitalWrite(kPinButton1, HIGH); //hidupkan pullup resistor
  digitalWrite(kPinButton2, HIGH); //hidupkan pullup resistor
}

void loop()
{
```

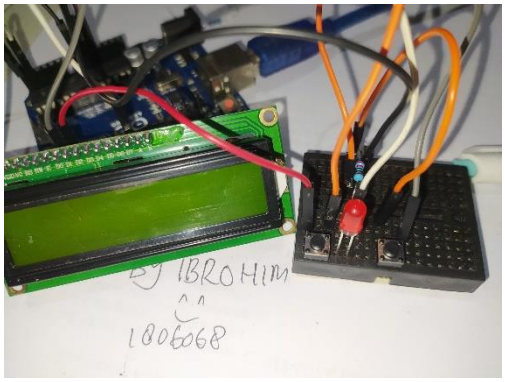
58

```
  if(digitalRead(kPinButton1) == LOW){
    ledBrightness--;
    }
  else if(digitalRead(kPinButton2) == LOW){
    ledBrightness++;
    }
  ledBrightness = constrain(ledBrightness,0,255);
  analogWrite(kPinLed, ledBrightness);
  delay(20);
  Serial.println(ledBrightness);
}
```

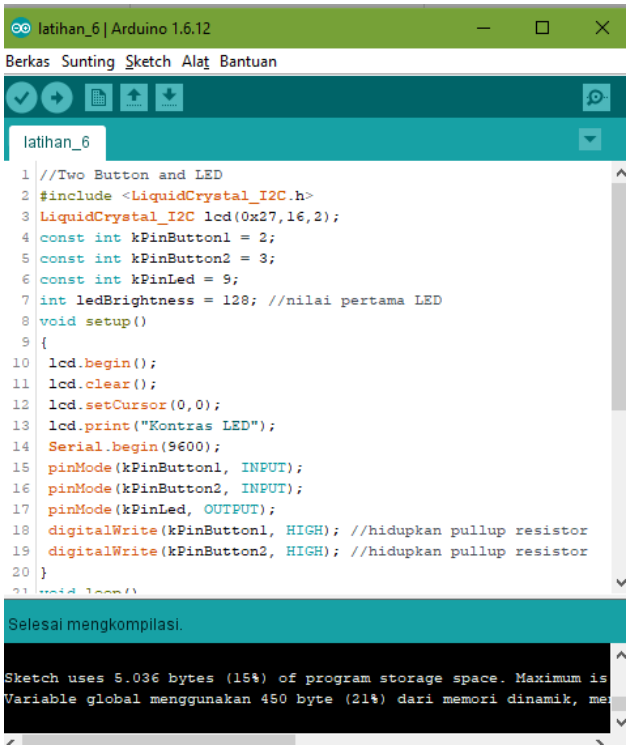
4. Amati apa yang terjadi pada praktikum ini dan buatlah kesimpulan.

Hasil Praktikum:

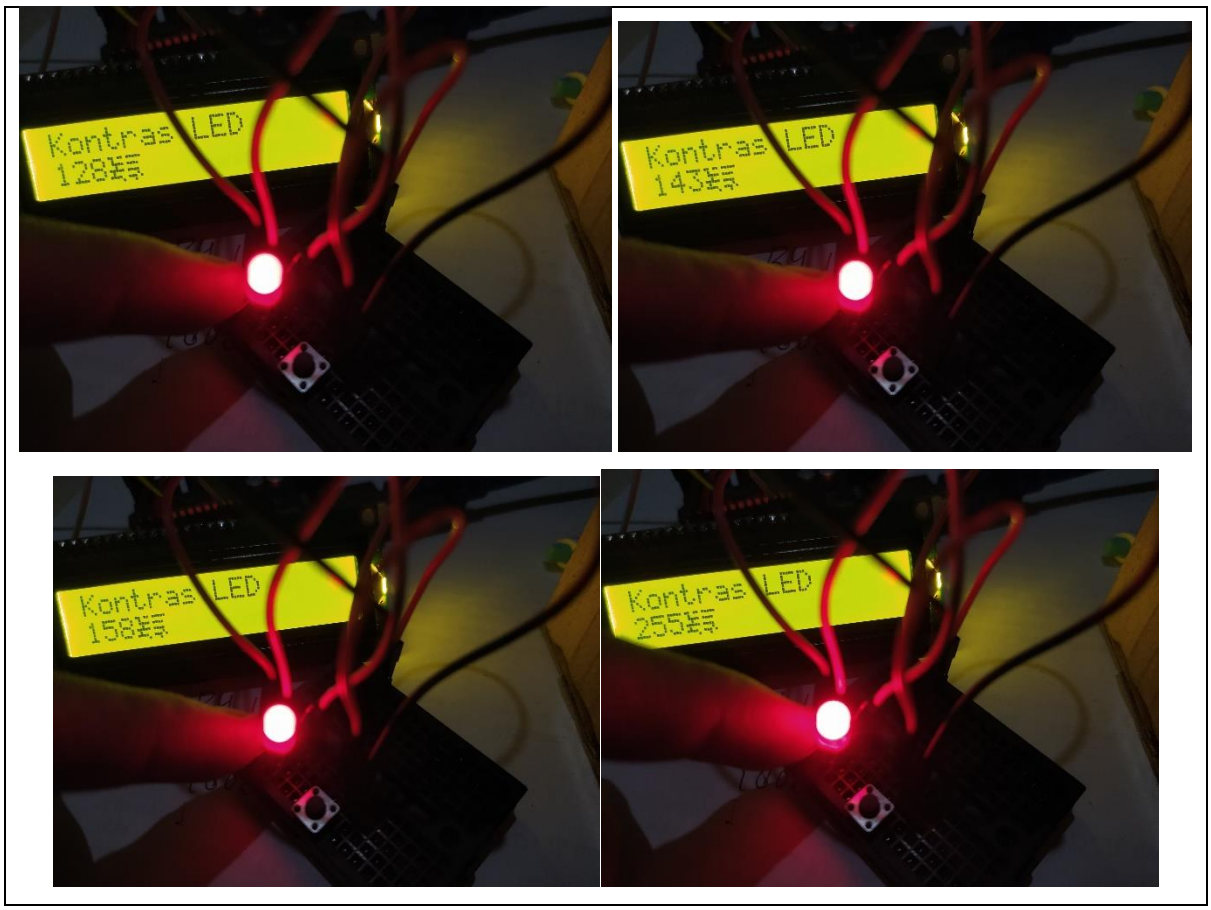
1. Hasil rangkaian pada breadboard.



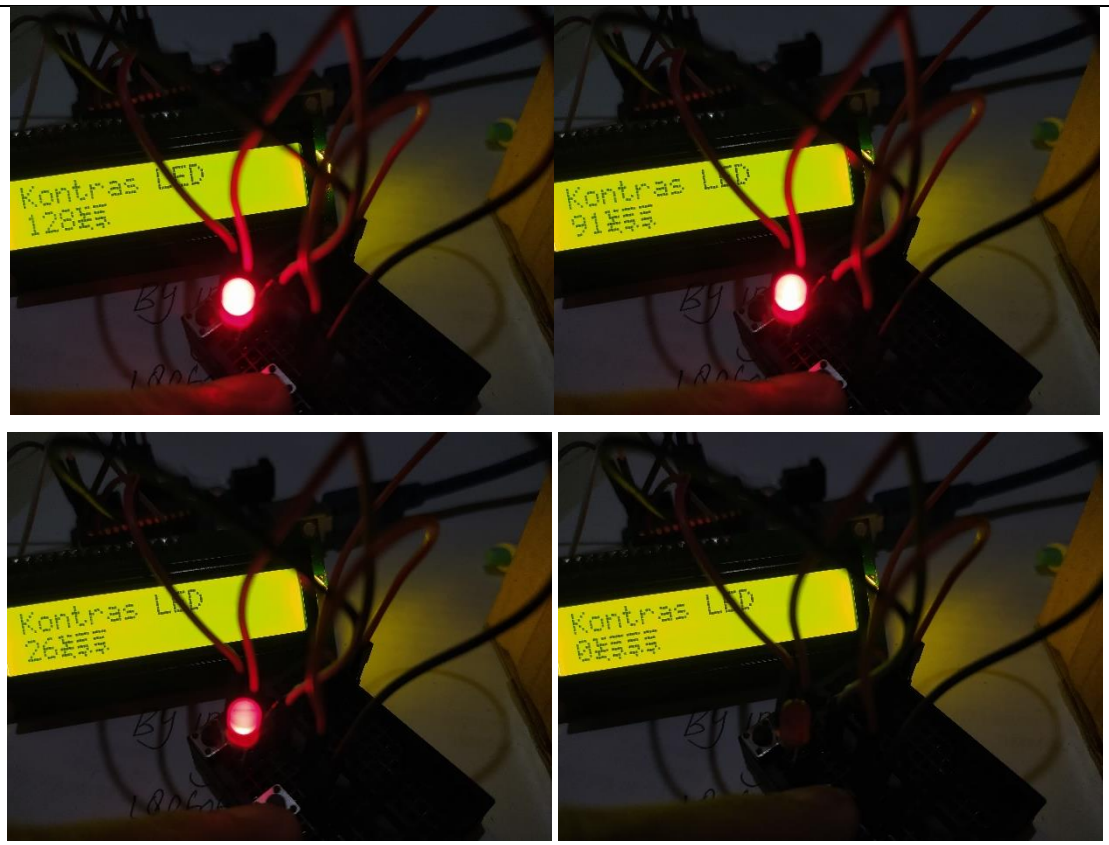
2. Hasil kode program yang sudah berhasil di-compile.



3. Hasil screenshot dokumentasi video praktikum saat kode program pada Arduino IDE berhasil di-upload ke board Arduino.



Saat pertama kali dinyalakan, led akan menyala dengan nilai kontras 128, setelah itu ditekan dengan pushbutton pwm naik nilai kontrassnya akan terus bertambah (sesuai lama menekan) sampai bernilai 255 (ditampilkan di LCD)



Hal yang sama juga berlaku sebaliknya untuk mengurangi kontrass, yakni setelah ditekan dengan pushbutton pwm turun nilai kontrassnya akan terus menurun (sesuai lama menekan) sampai bernilai 0 (ditampilkan di LCD).

Analisis

Pada latihan sebelumnya kita sudah mencoba menggunakan pushbutton sebagai switch, pada latihan kali ini kita akan memodifikasi program dan skematik dari program pushbutton. Latihan ini kita akan belajar cara mengatur tingkat kecerahan dari LED, program sebelumnya LED hanya menyala ketika posisi pin HIGH atau mati pada posisi LOW

Untuk cara pertama bisa digunakan dengan tambahan komponen yaitu potensiometer atau variable resistor, sedangkan cara kedua kita hanya memanfaatkan fungsi dari board Arduino yaitu fungsi PWM, pada Arduino UNO sudah dilengkapi dengan pin PWM, pin yang terdapat simbol ~ pada papan Arduino yaitu pin 3, 4, 5, 9, 10 dan 11. Kita dapatmemberikan rentan nilai 0 sampai 255. 0 berarti nilai tegangan pada pin tersebut bernilai 0V, sedangkan 255 tegangan pada pin tersebut bernilai 5v. Pada bahasa pemograman Arduino nilai tersebu dipanggil menggunakan fungsi analaogWrite().

Inisialisasi variable, dalam hal ini inisialisasi variable bernama kPinButton1 dengan type data konstanta integer dan nilai 2 (pada pin 2) untuk pin button 1, variable bernama kPinButton2 dengan type data konstanta integer dan nilai 3 untuk pinbutton 2, variable bernama kPinLed

dengan type data konstanta integer dan nilai 9 (pin pwm) untuk pin LED pwm, dan variable bernama ledBrightness dengan type data integer bernilai awal 128 untuk menampung nilai kecerahan LED.

```
//Two Button and LED
const int kPinButton1 = 2;
const int kPinButton2 = 3;
const int kPinLed = 9;
int ledBrightness = 128; //nilai pertama LED
```

Pada bagian selanjutnya yakni Void setup di baris ke-10 sampai ke-13 ditulis perintah konfigurasi LCD (tambahan menggunakan lcd). Lalu pada baris ke-14 terdapat perintah *Serial.begin(9600);* yang fungsinya untuk mengaktifkan serial monitor pada baudrate 9600. kemudian sebuah pin sebagai input atau ouput, dengan menggunakan perintah pinmode, pada variabel kPinButton1 dan kPinButton2 di set sebagai input dan pada bagian variabel kPinLed sebagai output. Serta bagian kPinButton1 dan kPinButton2 dihidupkan pullup resistor dengan perintah digitalWrite untuk mengatur nilai kontrassnya.

```
8  void setup()
9  {
10  lcd.begin();
11  lcd.clear();
12  lcd.setCursor(0,0);
13  lcd.print("Kontras LED");
14  Serial.begin(9600);
15  pinMode(kPinButton1, INPUT);
16  pinMode(kPinButton2, INPUT);
17  pinMode(kPinLed, OUTPUT);
18  digitalWrite(kPinButton1, HIGH); //hidupkan pullup resistor
19  digitalWrite(kPinButton2, HIGH); //hidupkan pullup resistor
```

Terakhir, pada bagian loop ditulis kode program yang akan dieksekusi terus menerus. Yang mana perintah pertama adalah masuk ke pengkodisian if dimana jika terbaca nilai kPinButton1 dengan perintah digitalRead bernilai LOW maka nilai variabel ledBrightness akan berkurang hingga 0, begitupun sebaliknya pada else if jika terbaca nilai kPinButton2 dengan perintah digitalRead bernilai LOW maka nilai variabel ledBrightness akan bertambah hingga 255 karena ada perintah constrain(ledBrightness,0,255). Lalu dengan perintah analogWrite(kPinLed, ledBrightness); maka akan diterapkan pada LEDnya. Lalu pada baris 32 dan 33 merupakan perintah menampilkan nilai kontras LED pada LCD, serta pada baris 34 merupakan perintah menampilkan LED pada serial monitor.

```
21 void loop()
22 {
23  if(digitalRead(kPinButton1) == LOW){
24  ledBrightness--;
25  }
26  else if(digitalRead(kPinButton2) == LOW){
27  ledBrightness++;
28  }
29  ledBrightness = constrain(ledBrightness,0,255);
30  analogWrite(kPinLed, ledBrightness);
31  delay(20);
32  lcd.setCursor(0,1);
33  lcd.println(ledBrightness);
34  Serial.println(ledBrightness);
35 }
```

Kesimpulan:

Berdasarkan praktikum Two Button and LED ini, kita dapat belajar cara mengatur tingkat kecerahan dari LED dengan memanfaatkan fungsi dari board Arduino yaitu fungsi PWM, pada Arduino UNO sudah dilengkapi dengan pin PWM, pin yang terdapat simbol ~ pada papan Arduino yaitu pin 3, 4, 5, 9, 10 dan 11. Kita dapatmemberikan rentan nilai 0 sampai 255. 0 berarti nilai tegangan pada pin tersebut bernilai 0V, sedangkan 255 tegangan pada pin tersebut bernilai 5v. Pada bahasa pemograman Arduino nilai tersebu dipanggil menggunakan fungsi analaogWrite().

Praktikum Sensor dan Sistem Mikroprosessor

Arduino Uno R3 Strater Kit

Latihan 7

Nama : Ibrohim
NIM : 1806068
Kelas : Elektronika Industri 2018

Judul : Light Sensor

Tujuan :

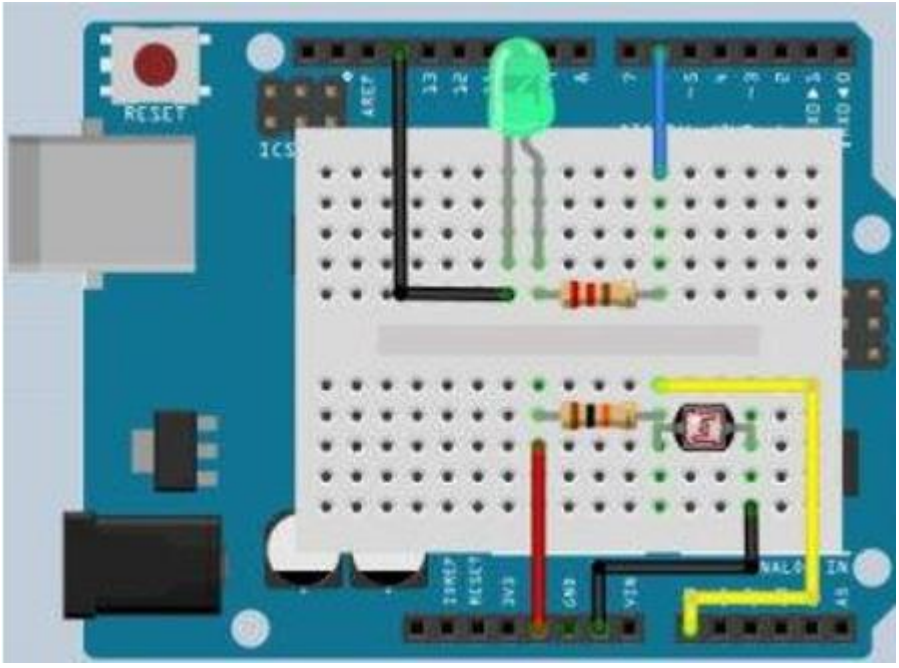
1. Menggunakan Arduino sebagai input.
2. Menggunakan Sensor LDR pada Arduino.
3. Menggunakan fungsi yang pin analog.
4. Membuat program sederhana Light Sensor pada lampu LED.

Alat dan Bahan :

1. Breadboard 1 buah
2. LED Merah 1 buah
3. Resistor 220 ohm 1 buah
4. Kabel jumper secukupnya
5. LDR 1 buah
6. Resistor 10k ohm 1 buah

Langkah Kerja:

1. Siapkan alat dan bahan
2. Buatlah rangkaian seperti di bawah ini pada breadboard.



Hubungkan seri LDR dan resistor 10K lalu diantara keduanya terhubung ke input analog A0, berikut merupakan sebuah pembagi tegangan, seperti gambar diatas, selanjutnya hubungkan LED kaki anoda ke pin digital 6 melalui resistor, dan kaki katoda ke ground.

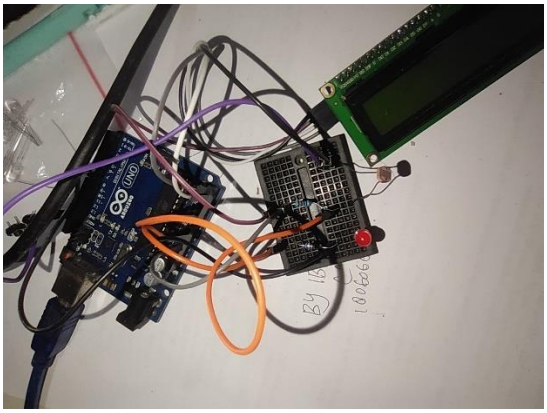
3. Compile dan upload kode program tersebut ke board Arduino Uno.

```
//Light sensor
int ledPin = 6; //pin LED
int ldrPin = 0; //pin LDR
int lightVal = 0; //nilai yang akan di baca LDR

void setup()
{
  pinMode(ledPin, OUTPUT);
}
void loop()
{
  //membaca nilai dari LDR
  lightVal = analogRead(ldrPin);
  digitalWrite(ledPin, HIGH); //Hidupkan LED
  //delay di tentukan dari pembacaan nilai LDR
  delay(lightVal);
  digitalWrite(ledPin, LOW); //Matikan LED
  delay(lightVal);
}
```

4. Amati apa yang terjadi pada praktikum ini dan buatlah kesimpulan.

Hasil Praktikum:

1. Hasil rangkaian pada breadboard.

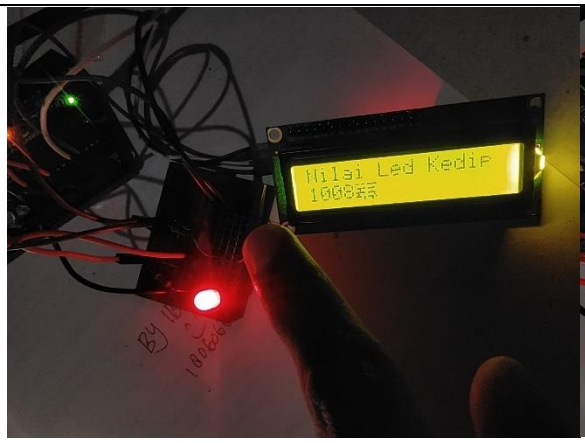


2. Hasil kode program yang sudah berhasil di-compile.

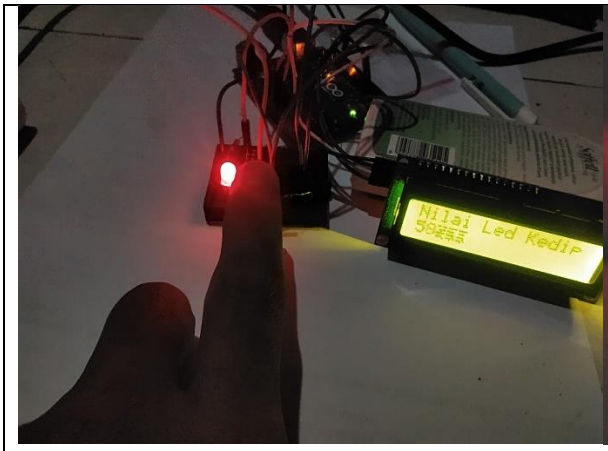
latihan_7

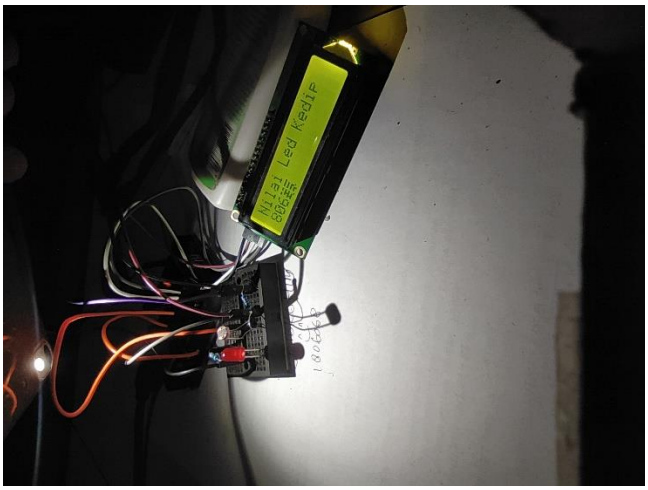
```
//Light Sensor
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);
int ledPin = 6; //pin LED
int ldrPin = 0; //pin LDR
int lightVal = 0; //nilai yang akan di baca LDR
void setup()
{
 lcd.begin();
 lcd.clear();
 lcd.setCursor(0, 0);
 lcd.print("Nilai Led Kedip");
 pinMode(ledPin, OUTPUT);
}
void loop()
{
 //membaca nilai dari LDR
 lightVal = analogRead(ldrPin);
 digitalWrite(ledPin, HIGH); //Hidupkan LED
 //delay di tentukan dari pembacaan nilai LDR
 delay(lightVal);
 digitalWrite(ledPin, LOW); //Matikan LED
 lcd.setCursor(0, 1);
 lcd.println(lightVal);
 delay(lightVal);
}
```

3. Hasil screenshot dokumentasi video praktikum saat kode program pada Arduino IDE berhasil di-upload ke board Arduino.



Pada konfigurasi rangkaian pertama (seri LDR dan resistor 10K lalu diantara keduanya terhubung ke input analog A), saat LDR pada ruangan gelap, maka nilai kontrasnya menjadi besar (971), lalu semakin digelapkan lagi (LDR ditutup dengan tangan) semakin besar nilainya (1008), dan saat di beri cahaya nilsinya menjadi semakin kecil (270). Oleh karena itu senakin terang, semakin kecil nilainya dan semakin cepat kedipan LEDnya.





Pada konfigurasi rangkaian kedua (posisi LDR ditukar dengan resistor 10k, posisi kaki LDR langsung ke supply sedangkan kaki resistor terhubung ke ground), saat LDR pada ruangan gelap, maka nilai kontrasnya menjadi kecil (104), lalu semakin digelapkan lagi (LDR ditutup dengan tangan) semakin keci; nilainya (58), dan saat di beri cahaya nilsinya menjadi semakin besar (806). Oleh karena itu senakin terang, semakin besar nilainya dan semakin lambat kedipan LEDnya.

Analisis

Pada latihan ini kita akan menggunakan LDR (Light Dependent Resistor) untuk membaca nilainya dan merubah kecepatan kedipan pada led.

Inisialisasi variable, dalam hal ini inisialisasi variable bernama ledPi dengan type data integer (pada pin 6) untuk pin led, variable bernama ldrPin dengan type data integer pada pin 0 (analog) untuk LDR, variable bernama lightVal dan dengan type data integer dan nilai 0 untuk menampung nilai LDR.

```
1 //Light Sensor
2 #include <LiquidCrystal_I2C.h>
3 LiquidCrystal_I2C lcd(0x27,16,2);
4 int ledPin = 6; //pin LED
5 int ldrPin = 0; //pin LDR
6 int lightVal = 0; //nilai yang akan di l
```

Pada bagian selanjutnya yakni Void setup di baris ke-7 sampai ke-12 ditulis perintah konfigurasi LCD (tambahan menggunakan lcd). kemudian sebuah pin sebagai input atau ouput, dengan menggunakan perintah pinmode, pada variabel ledPin di set sebagai sebagai output.

```
7  void setup()
8  {
9   lcd.begin();
10  lcd.clear();
11  lcd.setCursor(0, 0);
12  lcd.print("Nilai Led Kedip");
13  pinMode(ledPin, OUTPUT);
14 }
```

Terakhir, pada bagian loop ditulis kode program yang akan dieksekusi terus menerus. Yang mana perintah pertama adalah analogRead pada ldrPin yang kemudian dimasukan ke variabel lightVal. Lalu berikutnya perintah digitalWrite yang mengeset ledPin ke high, sehingga LED akan menyala. Dan perintah delay(lightVal) untuk mengeset delay sesui dengan nilai LDRnya. Serta prtintah digitalWrite(ledPin, LOW) agar meMmtikan LED. Lalu pada baris 23 sampai 25 merupakan perintah menampilkan nilai kontras LED pada LCD, serta pada baris 34 merupakan perintah menampilkan LED pada serial monitor.

```
15 void loop()
16 {
17  //membaca nilai dari LDR
18  lightVal = analogRead(ldrPin);
19  digitalWrite(ledPin, HIGH); //Hidupkan LED
20  //delay di tentukan dari pembacaan nilai LDR
21  delay(lightVal);
22  digitalWrite(ledPin, LOW); //Matikan LED
23  lcd.setCursor(0, 1);
24  lcd.println(lightVal);
25  delay(lightVal);
26 }
```

Kesimpulan:

Berdasarkan praktikum light sensor kita akan melihat LED berkedip on dan off. Pada keadaan pertama () Jika kita tutupi LDR (dalam keadaan gelap) maka LED akan berkedip lebih lambat dari sebelumnya dan jika kita terangi LDR menggunakan lampu maka LED akan terlihat berkedip lebih cepat dari sebelumnya. Sedangkan pada keadaan kedua yakni tukar posisi LDR dengan resistor 10k, posisi kaki LDR langsung ke supply sedangkan kaki resistor terhubung ke ground. Pada keadaan kedua hasilnya sebaliknya dengan keadaan pertama. Dengan begitu kita bisa melakukan konfigurasi LDR pada arduino hanya dengan memanfaatkan posisi penempatan LDRnya (keadaan 1 atau 2).

Praktikum Sensor dan Sistem Mikroprosessor

Arduino Uno R3 Strater Kit

Latihan 8

Nama : Ibrohim
NIM : 1806068
Kelas : Elektronika Industri 2018

Judul : Potensiometer

Tujuan :

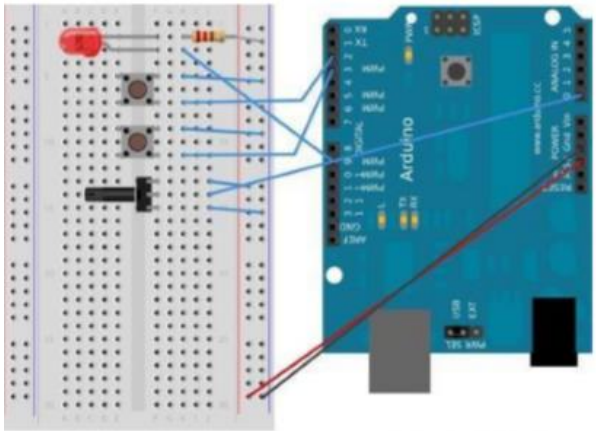
1. Menggunakan Arduino sebagai input.
2. Menggunakan potensiometer pada Arduino.
3. Menggunakan fungsi pin PWM pada arduino.
4. Membuat program sederhana Potensiometer untuk mengontrol pada lampu LED.

Alat dan Bahan :

1. Breadboard 1 buah
2. LED Merah 1 buah
3. Resistor 220 ohm 1 buah
4. Kabel jumper secukupnya
5. Potensiometer 10k 1 buah

Langkah Kerja:

1. Siapkan alat dan bahan
2. Buatlah rangkaian seperti di bawah ini pada breadboard.



Hubungkan kaki anoda led ke pin 9 dan kaki katoda LED ke kaki resitor yang dilanjutkan ke VCC, dan pin tengah potensiometer ke pin Analog 0 (A0).

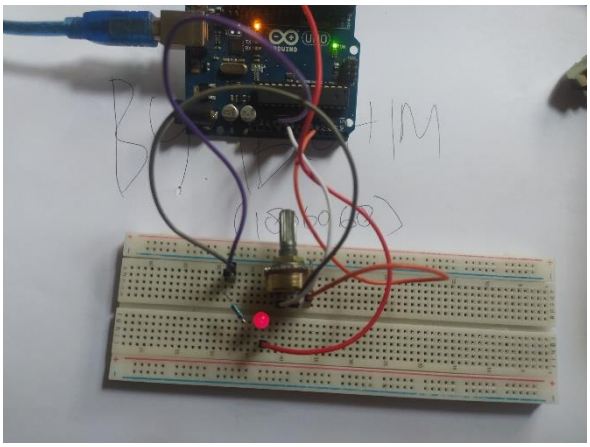
3. Compile dan upload kode program tersebut ke board Arduino Uno.

```
//Potensiometer
const int PinPot = A0;
const int PinLed = 9;
void setup()
{
  pinMode(PinPot, INPUT);
  pinMode(PinLed, OUTPUT);
}
void loop()
{
  int ledBrightness;
  int sensorValue = 0;
  sensorValue = analogRead(PinPot);
  ledBrightness = map(sensorValue,0,1023,0,255);
  analogWrite(PinLed, ledBrightness);
}
```

4. Amati apa yang terjadi pada praktikum ini dan buatlah kesimpulan.

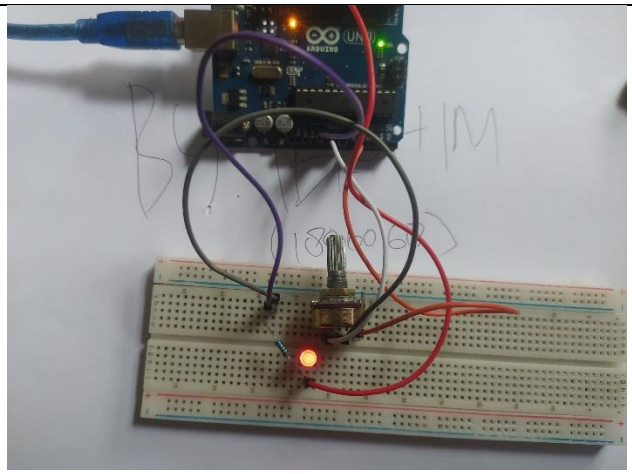
Hasil Praktikum:

1. Hasil rangkaian pada breadboard.

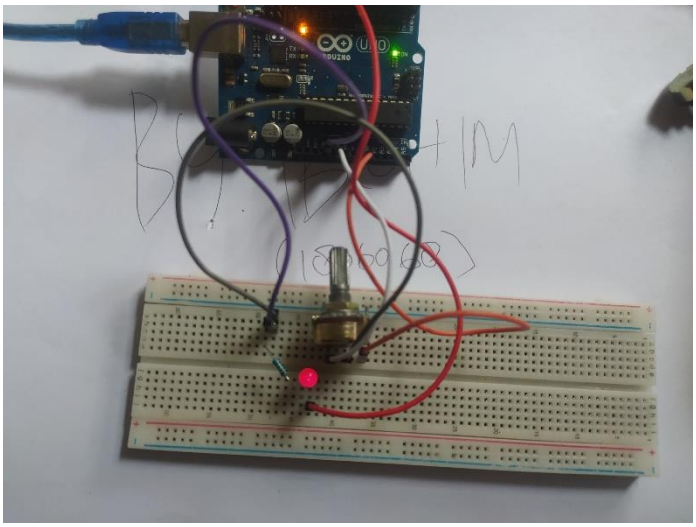


2. Hasil kode program yang sudah berhasil di-compile.

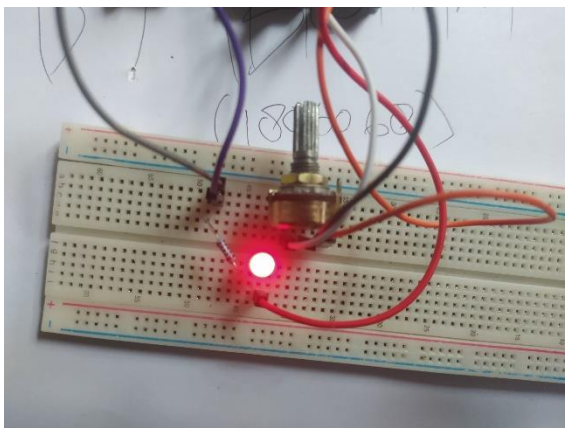
3. Hasil screenshot dokumentasi video praktikum saat kode program pada Arduino IDE berhasil di-upload ke board Arduino.



Pada kondisi awal seperti ini



Lalu saat diputar potensiometer berlawanan jarum jam, led semakin redup.



Dan pada saat potensiometer diputar searah jarum jam, LED semakin terang.

Analisis

Jika tadi kita menggunakan pushbutton maka pada latihan kali ini kita akan menggunakan potensio sebagai pengatur nilai PWM. Potensio merupakan sebuah resistor yang nilainya bisa diatur berdasarkan putaran pada knob potensio, nilai yang dihasilkan bisa bervariasi tergantung nilai maksimum yang tertera pada potensio, contoh jika nilai yang tertera

pada potensio bernilai 10k, maka itu merupakan nilai maximum dari sebuah potensio, dengan kata lain jika potensio diputar berlawan arah nilai maximum maka nilai akan berkurang sampai menuju 0 ohm.

Inisialisasi variable, dalam hal ini inisialisasi variable bernama PinPot dengan type data konstanta integer (pada pin A0) untuk pin potensiometer, variable bernama PinLed dengan type data konstanta integer pada pin 9 (analog) untuk LED.

```
1 //Potensiometer
2 const int PinPot = A0;
3 const int PinLed = 9;
```

Pada bagian selanjutnya yakni Void setup di baris ke-4 sampai ke-8 ditulis perintah k sebuah pin sebagai input atau ouput, dengan menggunakan perintah pinmode, pada variabel ledPin di set sebagai sebagai output. Dan PinPot diset sebagai input.

```
5 {
6  pinMode(PinPot, INPU
7  pinMode(PinLed, OUTP
8 }
```

Terakhir, pada bagian loop ditulis kode program yang akan dieksekusi terus menerus. Yang mana perintah pertama adalah *int ledBrightness;* untuk mensmpung variabel kecerahan LED, *int sensorValue = 0;* untuk menngeset nilai 0. Lalu perintah *sensorValue = analogRead(PinPot);* untuk mengeset nilai senseorValue bernilai seperti variabel PinPot. Lalu *ledBrightness = map(sensorValue,0,1023,0,255);* untuk menerapkan nilai sensorVallue ke ledBrightness sehingga kecerahan led akan mengikuti nilai pada potensiometer.

```
9  void loop()
10 {
11  int ledBrightness;
12  int sensorValue = 0;
13  sensorValue = analogRead(PinPot);
14  ledBrightness = map(sensorValue,0,1023,0,255);
15  analogWrite(PinLed, ledBrightness);
16 }
```

Pada code diatas terdapat code map(sensorValue, 0, 1023, 0, 255); kode tersebut mengubah nilai 0 sampai 1023 ke 0 sampai 255, berikut fungsi dari code diatas map(value, fromLow, fromHigh, toLow, toHigh) mengapa dirubah ? karena pada pin PWM hanya membaca nilai dari rentan 0 sampai 255, sedangkan pada input analog A0 bisa membaca nilai dari 0 sampai 1023.

Kesimpulan:

Berdasarkan praktikum potensiometer kita bisa membuat kontrol kecerahan LED dengan menggunkan potensiometer pada arduino. Selain itu juga kita dapat mengetahui kegunaan pin PWM saat bekerja dengan potensiometer 10k, sehingga bisa difungsikan sebagai kontrol LED dengan hanya menggunakan kode program yang sederhana seperti diatas. Dengan begitu kita menjadi lebih mudah dalam menerapkanya ke rangkaian yang lebih kompleks,

Praktikum Sensor dan Sistem Mikroprosessor

Arduino Uno R3 Strater Kit

Latihan 9

Nama : Ibrohim
NIM : 1806068
Kelas : Elektronika Industri 2018

Judul : Servo

Tujuan:

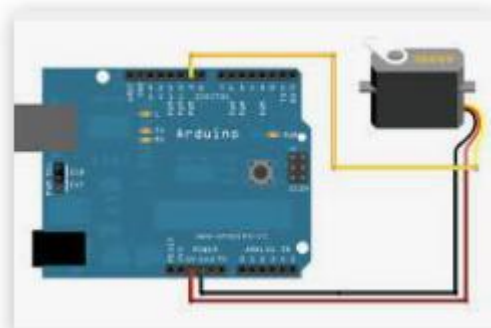
1. Menggunakan Arduino sebagai input.
2. Menerapkan konfigurasi pin servo pada arduino.
3. Melalukan konfigurasi agar servo berputar 180 derajat.
4. Membuat program sederhana dengan motor servo.

Alat dan Bahan :

1. Breadboard 1 buah
2. Servo SG9g/90 1 buah
3. Kabel jumper secukupnya

Langkah Kerja:

1. Siapkan alat dan bahan
2. Buatlah rangkaian seperti di bawah ini pada breadboard.



Pada servo umumnya terdapat 3 kabel yaitu kabel supply, ground dan data, warna kabel setiap servo terkadang berbeda, tapi pada kali ini kita akan menggunakan motor servo Merk Toper Pro 90/9g, pada servo ini kabel warna coklat untuk ground, merah untuk supply dan orange untuk data, hubungkan ketiga kabel tersebut ke arduino seperti pada gambar. Pada program kali ini kita akan memutar kondisi servo dari 0 derajat sampai 180 derajat, kanan dan kiri, begitu seterusnya.

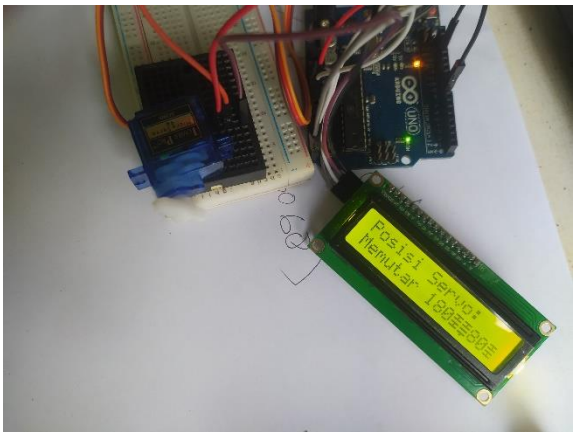
3. Compile dan upload kode program tersebut ke board Arduino Uno.

```
//Servo Sweep
#include <Servo.h>
Servo myservo;  //definisikan variabel myservo
sebagai servo

int pos = 0;    //variabel untuk memposisikan serv

void setup() {
  myservo.attach(8); //pin servo
}

void loop() {
  /*mulai dari step 0 derajat sampai step 180
derajat, setiap step bertambah 1*/
  for (pos = 0; pos <= 180; pos += 1) {
    //tulis nilai variable 'pos' pada servo
    myservo.write(pos);
    delay(15);
  }
   /*mulai dari step 180 derajat sampai step 0
derajat, setiap step berkurang 1*/
  for (pos = 180; pos >= 0; pos -= 1) {
    //tulis nilai variable 'pos' pada servo
    myservo.write(pos);
    delay(15);
  }
}
```

4. Amati apa yang terjadi pada praktikum ini dan buatlah kesimpulan.

Hasil Praktikum:

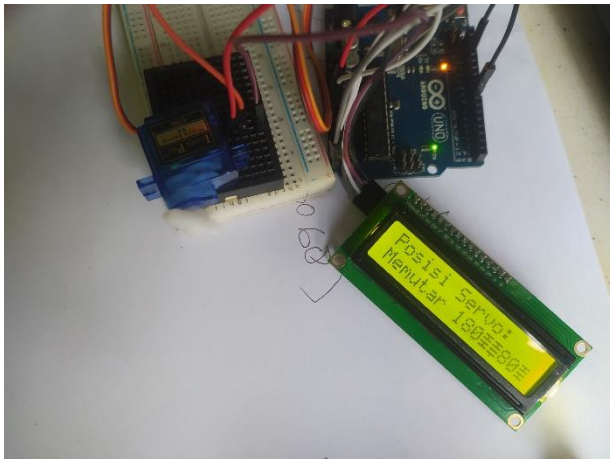
1. Hasil rangkaian pada breadboard.



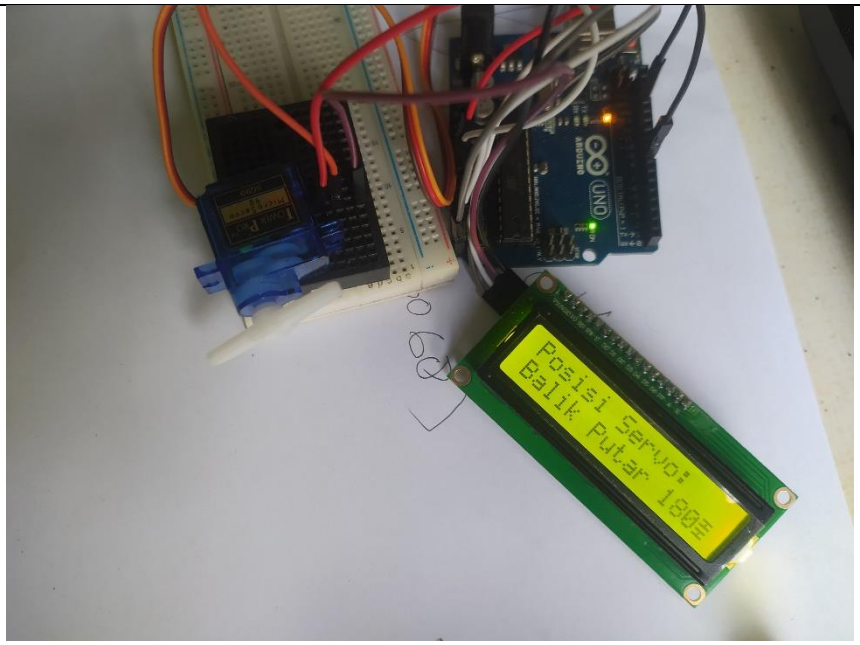
2. Hasil kode program yang sudah berhasil di-compile.

```
LiquidCrystal_I2C lcd(0x27,16,2);
#include <Servo.h>
Servo myservo; //definisikan variabel myservo sebagai servo
int pos = 0; //variabel untuk memposisikan servo
void setup() {
 lcd.begin();
 lcd.clear();
 lcd.setCursor(0, 0);
 lcd.print("Posisi Servo:");
 myservo.attach(8); //pin servo
}
void loop() {
 /*mulai dari step 0 derajat sampai step 180
derajat, setiap step bertambah 1*/
 for (pos = 0; pos <= 180; pos += 1) {
//tulis nilai variable 'pos' pada servo
 myservo.write(pos);
 lcd.setCursor(0, 1);
 lcd.println("Memutar 180");
 delay(15);
 }
 /*mulai dari step 180 derajat sampai step 0
derajat, setiap step berkurang 1*/
 for (pos = 180; pos >= 0; pos -= 1) {
//tulis nilai variable 'pos' pada servo
 myservo.write(pos);
 lcd.setCursor(0, 1);
 lcd.println("Balik Putar 180");
 delay(15);
 }
```

3. Hasil screenshot dokumentasi video praktikum saat kode program pada Arduino IDE berhasil di-upload ke board Arduino.



Kondisi saat sevo memutar 180 derajat



Lalu setelah itu sevo akan balik memutar 180 derajat lagi.

Analisis

Para projek kali ini kita akan menggunakan motor servo sebagai aktuator, hal pertama yang perlu kita ketahui adalah bagaimana cara menggerakan motor servo menggunakan arduino. Dimana i kita akan memutar kondisi servo dari 0 derajat sampai 180 derajat, kanan dan kiri, begitu seterusnya.

Pertama kita memasukan library servo.h dan library LCD sebagai indikatornya. Lalu kita Inisialisasi variable, dalam hal ini inisialisasi variable bernama pos dengan type data integer dan nilai 0 untuk nilai servo.

```
//Servo Sweep
#include <Servo.h>
Servo myservo;  //definisikan variabel myservo
sebagai servo

int pos = 0;    //variabel untuk memposisikan servo
```

Pada bagian selanjutnya yakni Void setup di baris ke-7 sampai ke-13 ditulis perintah konfigurasi LCD (tambahan menggunakan lcd). Lalu pada baris ke-12 terdapat perintah sebuah pin sebagai input atau ouput, dengan menggunakan perintah *myservo.attach(8);*, untuk mengeset pin servo (bagian input) ke pin 8.

```
7  void setup() {
8   lcd.begin();
9   lcd.clear();
10  lcd.setCursor(0, 0);
11  lcd.print("Posisi Servo:");
12  myservo.attach(8); //pin servo
13 }
```

Terakhir, pada bagian loop ditulis kode program yang akan dieksekusi terus menerus. Yang mana perintah pertama adalah masuk ke perulanagn for dimana perintah perulangan

pertama ini akan menerapkan servo berputar 180 derajat (searah jarum jam) pada baris ke 7 – 9. lalu menampilkan statusnya ke LCD (baris 10 – 11). Dan pada perintah 15 merupakan delay 12 milisekon.

```
4 void loop() {
5  /*mulai dari step 0 derajat sampai step 180
6 derajat, setiap step bertambah 1*/
7  for (pos = 0; pos <= 180; pos += 1) {
8 //tulis nilai variable 'pos' pada servo
9  myservo.write(pos);
0  lcd.setCursor(0, 1);
1  lcd.println("Memutar 180");
2  delay(15);
3  }
```

Lalu jika perintah diatas sudah selesai, akan masuk ke perintah perulangan berikutnya. Dimana pada perintah ini perulanganya akan diset ke kondisi kebalikan 180 derajat (baris 26 sampai 28). Lalu pada baris ke 29-30 merupakan perintah untuk agar status sevo ditampilkan di LCD. Dan baris ke 31 adalah perintah delay 15 milisekon.

```
24  /*mulai dari step 180 derajat sampai step 0
25 derajat, setiap step berkurang 1*/
26  for (pos = 180; pos >= 0; pos -= 1) {
27 //tulis nilai variable 'pos' pada servo
28  myservo.write(pos);
29  lcd.setCursor(0, 1);
30  lcd.println("Balik Putar 180");
31  delay(15);
32  }
33 }
```

Kesimpulan:

Berdasarkan praktikum Servo ini, kita dapat belajar cara menggunakan motor servo sebagai aktuator kususnya pada arduino. Dengan hanya menggunakan kode program perulangan yang sederhana, Kita dapat memutar kondisi servo dari 0 derajat sampai 180 derajat, kanan dan kiri, begitu seterusnya. Sehingga untuk penerapan pada rangkaian yang kompleks, kita bisa dengan mudah menerapkannya.

Praktikum Sensor dan Sistem Mikroprosessor

Arduino Uno R3 Strater Kit

Latihan 10

Nama : Ibrohim
NIM : 1806068
Kelas : Elektronika Industri 2018

Judul : Kontrol Servo

Tujuan:

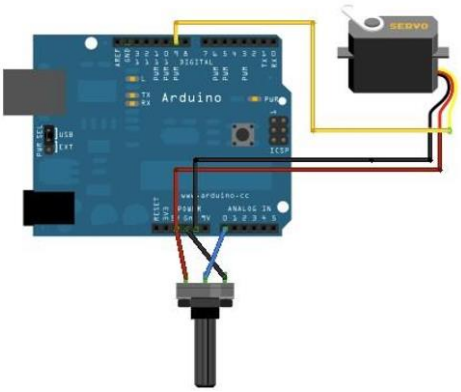
1. Menggunakan Arduino sebagai input.
2. Menerapkan konfigurasi pin servo pada arduino.
3. Melalukan konfigurasi agar servo berputar sesui putran potensiometer
4. Membuat program sederhana Kontrol Motor Servo sehingga bisa dikontrol lewat potensiometer.

Alat dan Bahan :

1. Breadboard 1 buah
2. Potensiometer 10k 1 buah
3. Servo SG9g/90 1 buah
4. Kabel jumper secukupnya

Langkah Kerja:

1. Siapkan alat dan bahan
2. Buatlah rangkaian seperti di bawah ini pada breadboard.



Hubungkan pin data servo ke pin 8, lalu hubungakan kaki 2 potensio (Posisi tengah) ke pin analog 0 arduino.

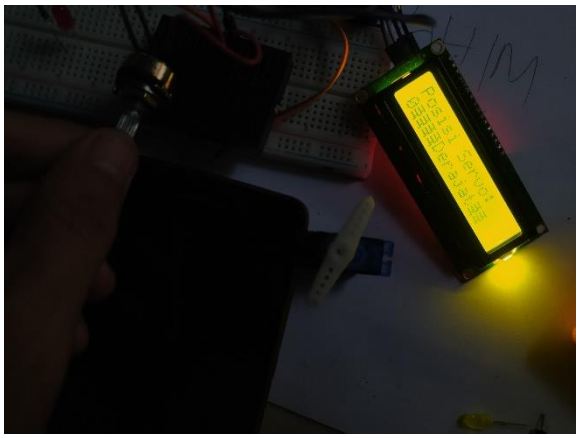
3. Compile dan upload kode program tersebut ke board Arduino Uno.

```
//Servo Control
#include <Servo.h>
Servo myservo;
int potpin = 0; //pin analog terkoneksi ke potensiometer
int val; //variabel untuk membaca nilai dari pin analog

void setup() {
      myservo.attach(8); //pin servo
}

void loop() {
      val = analogRead(potpin);
      //membaca nilai servo diantara 0 sampai 1023
      val = map(val, 0, 1023, 0, 180);
      //perubahan skala di perlukan untuk mengkonversi
      //daro 0 - 1023 ke 0 - 180
      myservo.write(val);
      delay(15);
}
```

4. Amati apa yang terjadi pada praktikum ini dan buatlah kesimpulan.

Hasil Praktikum:

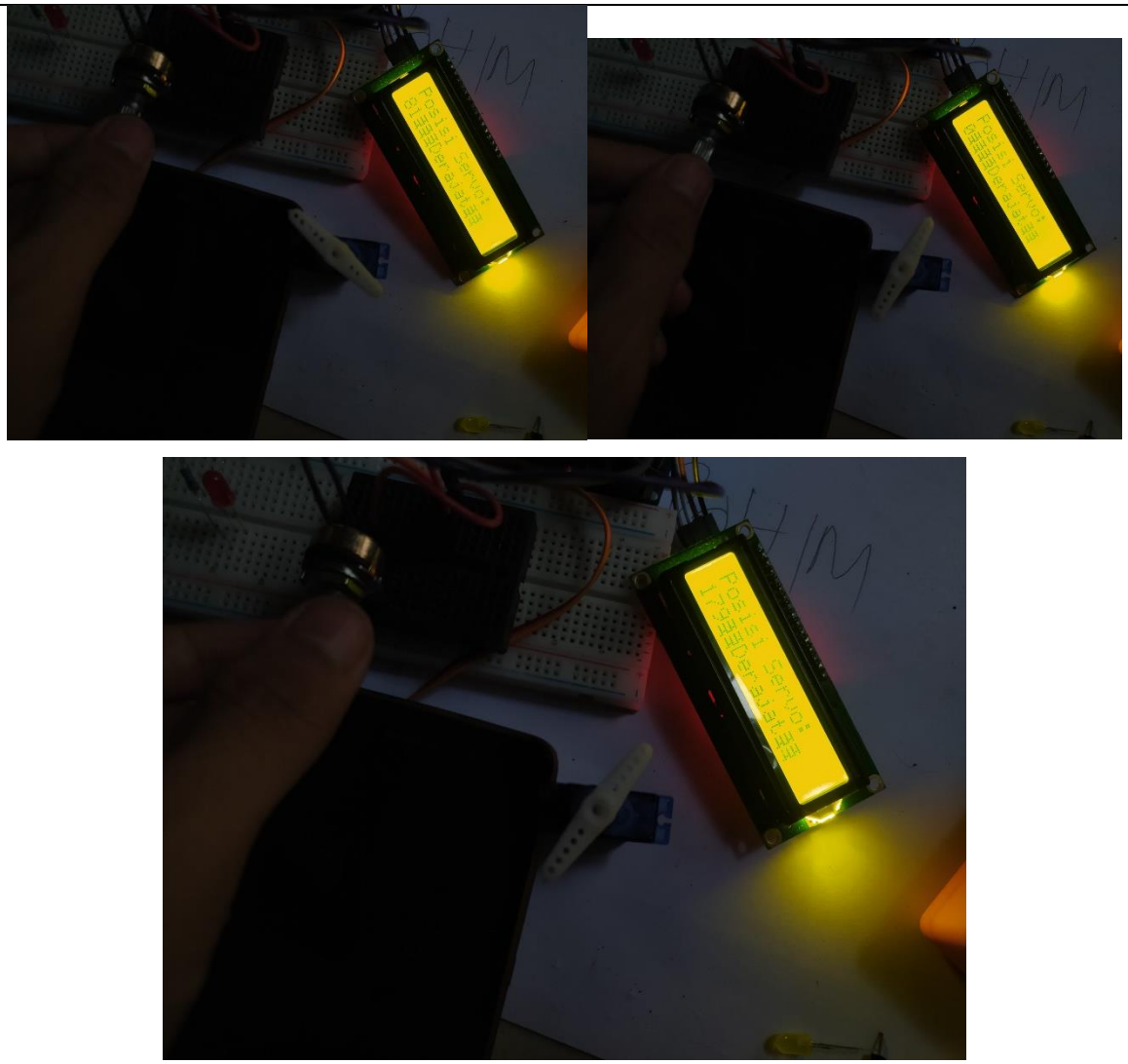
1. Hasil rangkaian pada breadboard.



2. Hasil kode program yang sudah berhasil di-compile.

```
lat_10
//Servo Control
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);
#include <Servo.h>
Servo myservo;
int potpin = 0; //pin analog terkoneksi ke potensiometer
int val; //variabel untuk membaca nilai dari pin analog
void setup() {
lcd.begin();
 lcd.clear();
 lcd.setCursor(0, 0);
 lcd.print("Posisi Servo:");
myservo.attach(8); //pin servo
}
void loop() {
val = analogRead(potpin);
//membaca nilai servo diantara 0 sampai 1023
val = map(val, 0, 1023, 0, 180);
//perubahan skala di perlukan untuk mengkonversi
//daro 0 - 1023 ke 0 - 180
myservo.write(val);
 lcd.setCursor(0, 1);
 lcd.println(val);
 lcd.setCursor(5, 1);
 lcd.println("Derajat");
delay(15);
}
```

3. Hasil screenshot dokumentasi video praktikum saat kode program pada Arduino IDE berhasil di-upload ke board Arduino.



Pada gambar tersebut, terlihat bahwasanya kita bisaz mengontrol servo sesui dengan gerakan potensiometer

Analisis

Latihan selanjutnya lanjutan dari latihan servo sebelumnya, akan tetapi jika pada latihan sebelumnya posisi servo sudah di atur untuk menempati posisi tertentu, kali ini kita akan mengatur posisi servo menggunakan potensiometer.

Pertama kita memasukan library servo.h dan library LCD sebagai indikatornya. Lalu kita Inisialisasi variable, dalam hal ini inisialisasi variable bernama potpin dengan type data integer dan nilai 0 untuk nilai potensiometer pada pin analog. Variabel val dengan type integer yang digunakan untuk membaca nilai dari pin analog.

lat_10

```
//Servo Control
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);
#include <Servo.h>
Servo myservo;
int potpin = 0; //pin analog terkoneksi ke potensiometer
int val; //variabel untuk membaca nilai dari pin analog
```

Pada bagian selanjutnya yakni Void setup di baris ke-8 sampai ke-12 ditulis perintah konfigurasi LCD (tambahan menggunakan lcd). Lalu pada baris ke-13 terdapat perintah sebuah pin sebagai input atau ouput, dengan menggunakan perintah *myservo.attach(8);*, untuk mengeset pin servo (bagian input) ke pin 8.

```
7  int val; //variabel untuk memba
8  void setup() {
9  lcd.begin();
10  lcd.clear();
11  lcd.setCursor(0, 0);
12  lcd.print("Posisi Servo:");
13 myservo.attach(8); //pin servo
14 }
```

Terakhir, pada bagian loop ditulis kode program yang akan dieksekusi terus menerus. Yang mana perintah pertama adalah *val = analogRead(potpin);* yang fungsinya untuk membaca nilai servo diantara 0 sampai 1023. Lalu perintah *val = map(val, 0, 1023, 0, 180);* yang fungsinya untuk melakukan perubahan skala di perlukan untuk mengkonversi dari 0 - 1023 ke 0 – 180. Dan perintah *myservo.write(val);* untuk menerapkan perintah sebelumny ke servo. Serta menampilkan nilainya ke LCD (baris 22 – 25). Dan pada baris 26 merupakan delay 15 milisekon.

```
15 void loop()
16 val = analogRead(potpin);
17 //membaca nilai servo diantara 0 sampai 1023
18 val = map(val, 0, 1023, 0, 180);
19 //perubahan skala di perlukan untuk mengkonversi
20 //daro 0 - 1023 ke 0 - 180
21 myservo.write(val);
22  lcd.setCursor(0, 1);
23  lcd.println(val);
24  lcd.setCursor(5, 1);
25  lcd.println("Derajat");
26 delay(15);
27 }
```

Kesimpulan:

Berdasarkan praktikum Servo ini, kita dapat belajar cara menggunakan potensiometer sebagai kontrol motor servo kususnya pada arduino. Dengan hanya menggunakan kode program yang sederhana, Kita dapat memutar kondisi servo sesuai dengan posisi pada potensimeter, kanan dan kiri, begitu seterusnya. Selain itu kita juga dapat mengerti fungsi PWM untuk merapkanya pada kontrol servo ini. Sehingga untuk penerapan pada rangkaian yang kompleks, kita bisa dengan mudah menerapkannya.

Praktikum Sensor dan Sistem Mikroprosessor

Arduino Uno R3 Strater Kit

Latihan 11

Nama : Ibrohim
NIM : 1806068
Kelas : Elektronika Industri 2018

Judul : LCD 16x2

Tujuan:

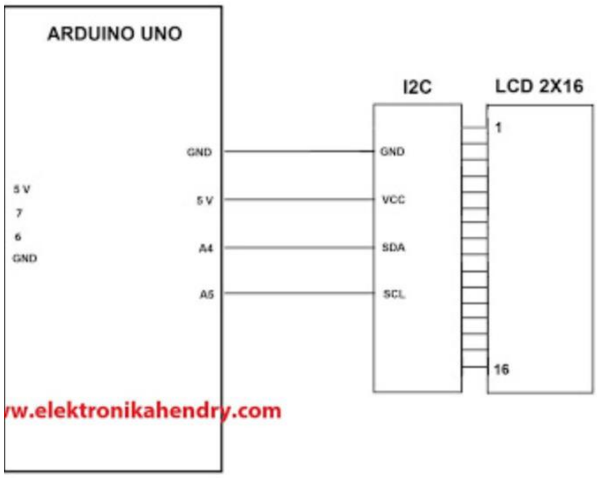
1. Menggunakan LCD pada Arduino.
2. Membuat program sederhana untuk menampilkan text pada LCD 16 x 2.
3. Membuat program sederhana untuk menampilkan karakter blink pada LCD 16 x 2.
4. Membuat program sederhana untuk agar bisa memasukan text ke serial input pada LCD 16 x 2.
5. Membuat program sederhana untuk menampilkan text autoscroll pada LCD 16 x 2.

Alat dan Bahan :

1. Breadboard 1 buah
2. LCD 16 x 2 1 buah
3. I2C LCD 1 buah
4. Kabel jumper secukupnya

Langkah Kerja:

1. Siapkan alat dan bahan
2. Buatlah rangkaian seperti di bawah ini pada breadboard.



Pertama gabungkan modul i2c dengan LCD dengan cara disolder, lalu pada LCD hubungkan pin SCL pada pin A5 arduino, SDA pada pin A4, GND pada pin GND, VCC pada pin 5V.

3. Compile dan upload kode program tersebut satu persatu (4 project) ke board Arduino Uno.

_1 §

```
//LCD 16x2 text "Hello World"

#include <LiquidCrystal_I2C.h>
//inisialisasi nomor pin LCD
LiquidCrystal_I2C lcd(0x27,16,2);
void setup() {
//set kolom dan baris
lcd.begin();
//tampilkan kata
lcd.print("hello, world!");
}
void loop() {
lcd.setCursor(0, 1);
//cursor di posisikan pada kolom 0 baris 1
lcd.print(millis() / 1000);
}
```

Berkas Sunting Sketch Alat Bantuan

_4 §

```
//Serial Input LCD i2c

#include <LiquidCrystal_I2C.h>
//inisialisasi nomor pin LCD
LiquidCrystal_I2C lcd(0x27,16,2);
void setup() {
 lcd.begin();
}
void loop() {
 //set kursor pada kolom dan baris 0
 lcd.setCursor(0, 0);
 // print dari 0 sampai 9:
 for (int thisChar = 0; thisChar < 10; thisChar++) {
 lcd.print(thisChar);
 delay(500);
 }
 //set kursor pada kolom 16 dan baris 1
 lcd.setCursor(16, 1);
 //set LCD untuk autoscroll
 lcd.autoscroll();
 // print dari 0 sampai 9:
```

_2 §

```
//LCD Blinking i2c

#include <LiquidCrystal_I2C.h>
//inisialisasi nomor pin LCD
LiquidCrystal_I2C lcd(0x27,16,2);
void setup() {
 //set kolom dan baris
 lcd.begin();
 // tampilkan kata
 lcd.print("hello, world!");
}
void loop() {
 //matikan cursor blinking
 lcd.noBlink();
 delay(3000);
 //hidupkan cursor blinking
 lcd.blink();
 delay(3000);
}
```

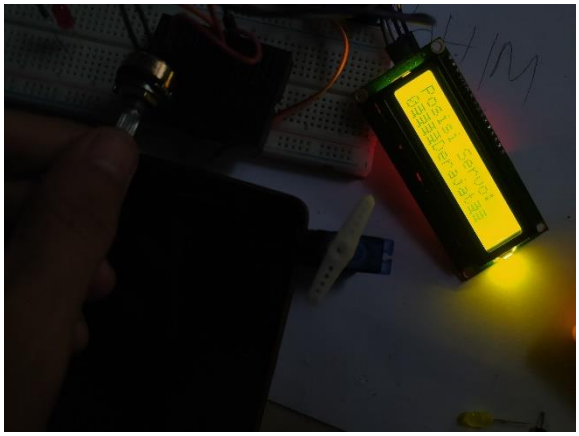
_3 §

```
//LCD Serial Input I2C

#include <LiquidCrystal_I2C.h>
//inisialisasi nomor pin LCD
LiquidCrystal_I2C lcd(0x27,16,2);
void setup() {
 lcd.begin();
 Serial.begin(9600);
}
void loop() {
 //jika karakter di terima oleh serial port
 if (Serial.available()) {
 //menunggu pesan masuk
 delay(100);
 lcd.clear(); //menghapus LCD
 // membaca semua karakter yang masuk
 while (Serial.available() > 0) {
//menulis ke LCD dari hasil bacaan serial yang masuk
 lcd.write(Serial.read());
 }
 }
```

4. Amati apa yang terjadi pada praktikum ini dan buatlah kesimpulan.

Hasil Praktikum:

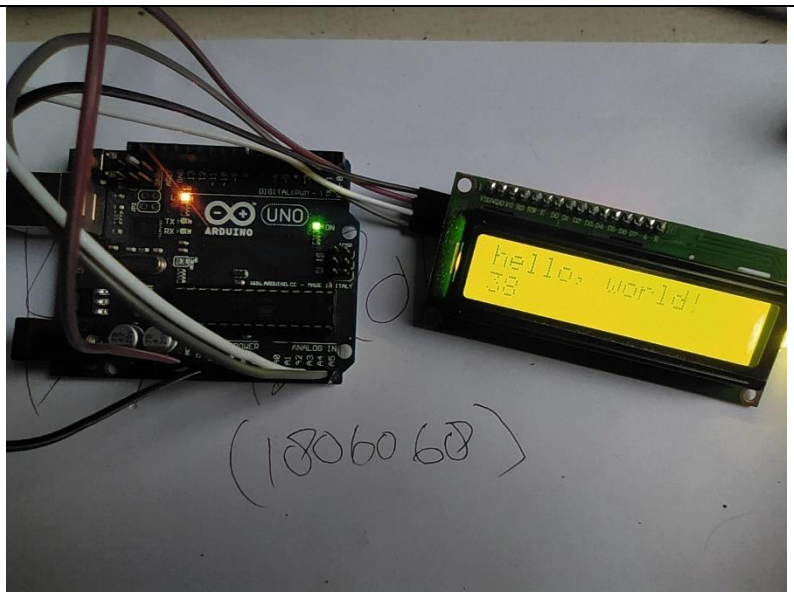
1. Hasil rangkaian pada breadboard.



2. Hasil kode program yang sudah berhasil di-compile.

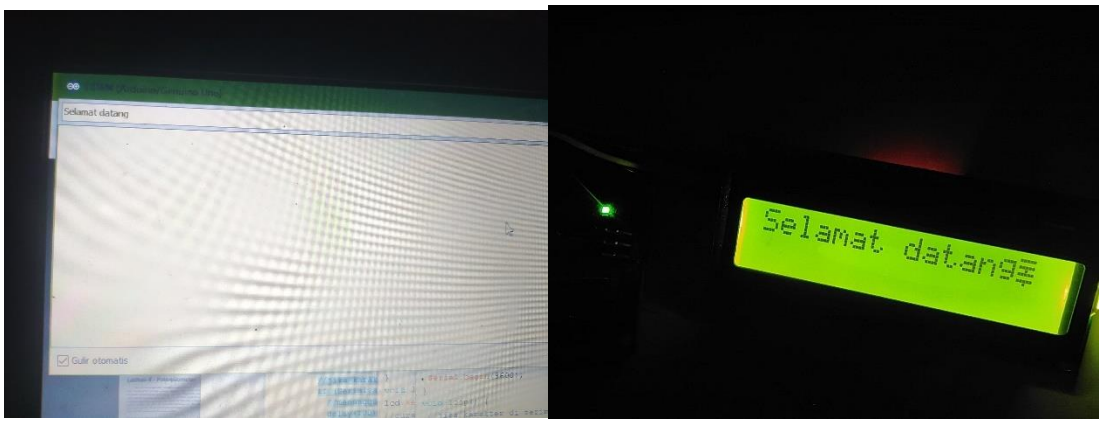
Gambar seperti pada langkah kerja ke-3.

3. Hasil screenshot dokumentasi video praktikum saat kode program pada Arduino IDE berhasil di-upload ke board Arduino.



Pada gambar tersebut, terlihat bahwasanya LCD menampilkan tulisan “Hello World!” seperti yang kita inginkan.

Pada gambar tersebut, terlihat bahwasanya LCD menampilkan tulisan “Hello World!” diikuti dengan karakter yang berkedip (blink) .



Pada gambar tersebut, terlihat bahwasanya dengan menggunakan fungsin interupt, kita bisa memasukan text pada serial monitor sehingga LCD menampilkan tulisan sesuai dengan yang ditulis pada serial monitor (interupt).

Pada beberapa gambar tersebut, terlihat bahwasanya kita LCD menampilakn text berjalan 0-9 dari atas kiri ke atas kekanan. Lalu diteruskan dari bawah kanan ke bawah kiri. Dan begitu seterusnya.

Analisis

Latihan kali ini adalah menggunakan LCD, tetapi disini saya menggunakan LCD dengan bantuan modul I2C, karena dengan modul ini bisa menghemat pin pada arduino, sehingga dapat digunakan pada project dengan banyak inputan.

Dalam latihan ini terdapat 4 latihan LCD yang menggunakan rangkaian yang sama dan seting awalan (librari dan variabel) yang sama (tanpa variabel). Dimana untuk rangkaianya ada pada langkah-langkah, dan untuk program awalnya adalah dengan cara input library lcd i2c berikut.

```
//LCD Serial Input
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
```

Setelah itu, berikut penjelasan masing-masing keempat kode programnya:

Menampilkan "Hello World!" dengan timer

```
6  void setup() {
7  //set kolom dan baris
8  lcd.begin();
9  //tampilkan kata
10 lcd.print("hello, world!");
11 }
12 void loop() {
13 lcd.setCursor(0, 1);
14 //cursor di posisikan pada kolom 0 baris 1
15 lcd.print(millis() / 1000);
16 }
```

Pada kode diatas, hasil yang di dapat yaitu LCD pada kolom 0 baris 0 akan menampilkan tulisan hello, world!, (karena kolom dan baris pertama dimulai dari angka 0, jadi kolom/baris 1 = kolom/baris 0, kolo/baris 2 = kolom/baris 1) dan di kolom pertama baris 1 akan menampilkan hitungan dalam detik. Seperti program lainya, LCD dijalankan menggunakan perintah perintah khusus yang sudah di definisikan di library LiquidCrystal_I2C.h seperti perintah lcd.print() untuk menampilkan data dan perintah lcd.setCursor(0,1) yaitu menempatkan kursor pada kolom 0 baris 1, berikut program-program LCD lainya. Untuk skematik sama seperti skematik program LCD.

Menampilkan "Hello World!" dengan blink karakter

```
void setup() {
  //set kolom dan baris
  lcd.begin(16, 2);
  // tampilkan kata
  lcd.print("hello, world!");
}

void loop() {
  //matikan cursor blinking
  lcd.noBlink();
  delay(3000);
  //hidupkan cursor blinking
  lcd.blink();
  delay(3000);
}
```

Pada kode diatas, hasil yang di dapat yaitu LCD pada kolom 0 baris 0 akan menampilkan tulisan hello, world!, (karena kolom dan baris pertama dimulai dari angka 0, jadi kolom/baris 1 = kolom/baris 0, kolo/baris 2 = kolom/baris 1) dan di kolom pertama baris 1

akan menampilkan karakter blink (kedip-kedip) yang bekedip 3 ditik dan mati 3 detik. Seperti program lainya, LCD dijalankan menggunakan perintah perintah khusus yang sudah di definisikan di library LiquidCrystal_I2C.h seperti perintah lcd.print() untuk menampilkan data dan perintah lcd.setCursor(0,1) yaitu menempatkan kursor pada kolom 0 baris 1, berikut program-program LCD lainya. Untuk skematik sama seperti skematik program LCD.

Menampilkan Text Sesuai dengan Input Serial Monitor (Interupt)

```
void setup() {
  lcd.begin(16, 2);
  Serial.begin(9600);
}
void loop() {
  //jika karakter di terima oleh serial port
  if (Serial.available()) {
    //menunggu pesan masuk
    delay(100);
    lcd.clear(); //menghapus LCD
  // membaca semua karakter yang masuk
  while (Serial.available() > 0) {
 //menulis ke LCD dari hasil bacaan serial yang masuk
    lcd.write(Serial.read());
    }
  }
}
```

Pada kode diatas, hasil yang di dapat yaitu LCD pada kolom 0 baris 0 akan menampilkan tulisan sesuai dengan yang dituliskan pada serial monitor baud rate 9600,(karena kolom dan baris pertama dimulai dari angka 0, jadi kolom/baris 1 = kolom/baris 0, kolo/baris 2 = kolom/baris 1). Seperti program lainya, LCD dijalankan menggunakan perintah perintah khusus yang sudah di definisikan di library LiquidCrystal_I2C.h seperti perintah lcd.print() untuk menampilkan data dan perintahperintah lain seperti pada gambar sehingga bisa membuat interupt. Untuk skematik sama seperti skematik program LCD.

Menampilkan Text Autoscrol (1 2 3 4 5 6 7 8 9 0)

```
void loop() {
  //set kursor pada kolom dan baris 0
  lcd.setCursor(0, 0);
  // print dari 0 sampai 9:
  for (int thisChar = 0; thisChar < 10; thisChar++)
    lcd.print(thisChar);
    delay(500);
  }

  //set kursor pada kolom 16 dan baris 1
  lcd.setCursor(16, 1);
  //set LCD untuk autoscroll
  lcd.autoscroll();
  // print dari 0 sampai 9:
  for (int thisChar = 0; thisChar < 10; thisChar++)
    lcd.print(thisChar);
    delay(500);
  }
  //matikan automatic scrolling
  lcd.noAutoscroll();
  //bersihkan LCD untuk loop selanjutnya
  lcd.clear();
}
```

Pada kode diatas, hasil yang di dapat yaitu LCD pada kolom 0 baris 0 akan menampilkan tulisan angka 1 2 3 4 5 6 7 8 9 0 (karena kolom dan baris pertama dimulai dari angka 0, jadi kolom/baris 1 = kolom/baris 0, kolo/baris 2 = kolom/baris 1) dimana angka tersebut ditulis menggunakan fungsi perulangan for sehingga bisa muncul satu persatu

angkanya dan berjalan scroll dari atas kiri ke atas kanan dan dilanjutkan dari kebawah kanan ke bawah kiri. Hal tersebut karena program diatas menggunakan fungsi lcd.autoscroll(); dan lcd.noAutoscroll(); yang merupakan perintah fungsi auto scroll pada library LiquidCrystal_I2C.h pada arduino

Kesimpulan:

Berdasarkan praktikum LCD ini, dengan bantuan modul i2c kita menjadi menghemat pin untuk lcd pada arduino. Selain itu dengan hanya satu rangkaian kita bisa membuat berbagai macam perintah pada lcd 16x2 karena adanya library LiquidCrystal_I2C.h diantaranya: menampilkan text dengan timer pada baris tertentu, membuat karakter blink, menggunakan interupt (masukan text sesuai yang kita inginkan), dan membuat autoscroll text. Selain itu banyak lagi fungsi lain yang ada pada library tersebut. Dimana dengan memahami contoh tersebut tentu kita dapat menerapkanya ke berbagai LCD project arduino yang lebih kompleks.

Praktikum Sensor dan Sistem Mikroprosessor

Arduino Uno R3 Strater Kit

Latihan 12

Nama : Ibrohim
NIM : 1806068
Kelas : Elektronika Industri 2018

Judul : Ultrasonic

Tujuan:

1. Menggunakan Arduino sebagai input.
2. Menggunakan Sensor Ultrasonik pada Arduino.
3. Membuat program sederhana dengan Sensor Ultrasonik pada arduino.

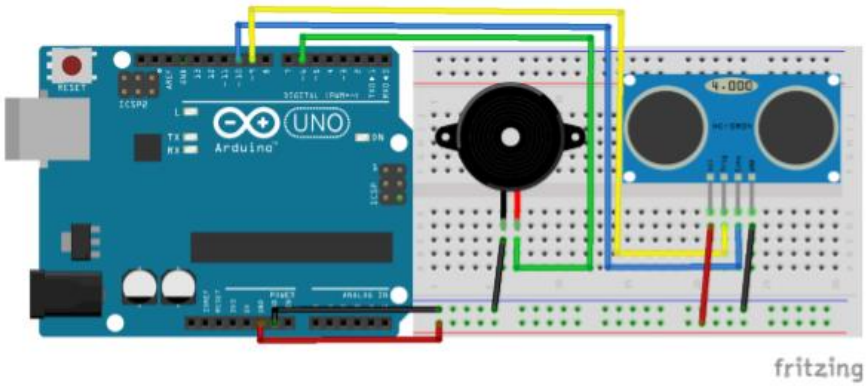
Alat dan Bahan :

1. Breadboard 1 buah
2. Sensor Ultrasonik 1 buah
3. Buzzer 1 buah
4. Kabel jumper secukupnya

Langkah Kerja:

1. Siapkan alat dan bahan
2. Buatlah rangkaian seperti di bawah ini pada breadboard.

Bagian 1



Bagian 2

Pada sensor Ultrasonik HC-SR04 memiliki 4 pin yaitu pin vcc, ground, trigger dan echo. Hubungkan pin vcc dan groung dari sensor ke arduino, pin trigger ke pin 9 dan pin echo ke pin 10 arduino.

```
const int trigPin = 9; //pin trigger
const int echoPin = 10; //pin echo
long duration;
int distance;

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
}

void loop() {
  // Clears pin trigger
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  //set pin trigger selama 10 microsecond
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  //membaca jarak yang dihasilkan
  duration = pulseIn(echoPin, HIGH);
  //memperkirakan jarak
  distance= (duration*0.034)/2;
  //tampilkan diserial monitor
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println("cm");
}
```

Bagian 1

lat_12_2

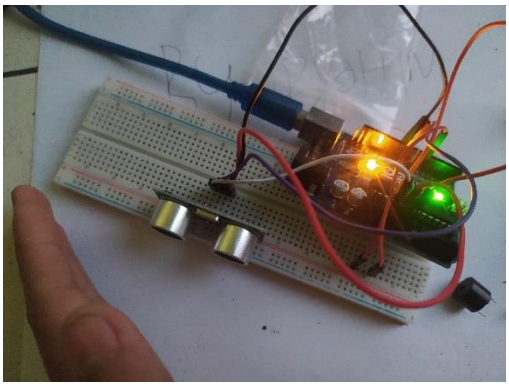
```
const int trigPin = 9; //pin trigger
const int echoPin = 10; //pin echo
int piezo=6; //Piezo speaker di pin 6.
long duration;
int distance;
void setup() {
 pinMode(trigPin, OUTPUT);
 pinMode(echoPin, INPUT);
 pinMode(piezo, OUTPUT);
 Serial.begin(9600);
}
void loop() {
 // Clears pin trigger
 digitalWrite(trigPin, LOW);
 delayMicroseconds(2);
 //set pin trigger selama 10 microsecond
 digitalWrite(trigPin, HIGH);
 delayMicroseconds(10);
 digitalWrite(trigPin, LOW);
 //membaca jarak yang dihasilkan
 duration = pulseIn(echoPin, HIGH);
 //memperkirakan jarak
 distance= (duration*0.034)/2;
 //tampilkan diserial monitor
 if (distance >= 200 || distance <= 0)
 {
 Serial.println("No measurement");
 }
 else
 {
```

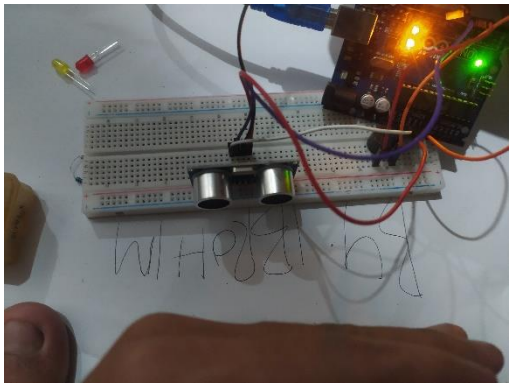
Bagian 2

3. Compile dan upload kode program tersebut ke board Arduino Uno.
4. Amati apa yang terjadi pada praktikum ini dan buatlah kesimpulan.

Hasil Praktikum:

1. Hasil rangkaian pada breadboard.





Bagian 1 Bagian 2

2. Hasil kode program yang sudah berhasil di-compile.

lat_12

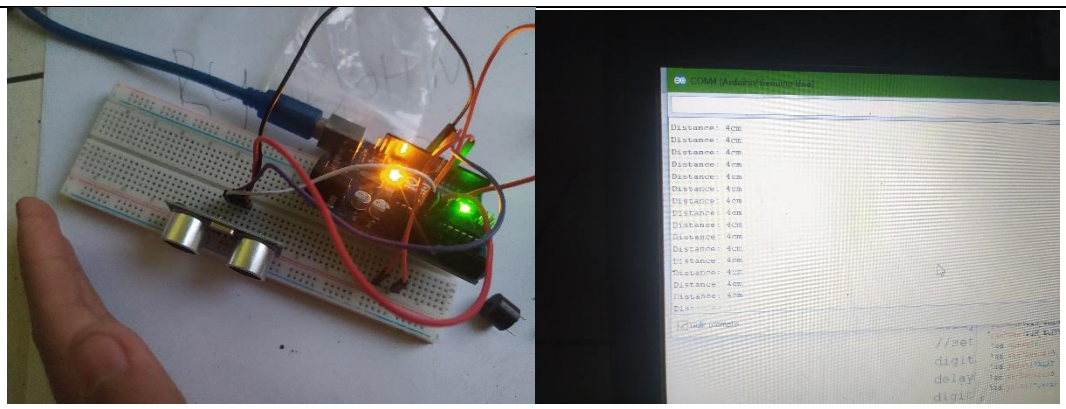
```
const int trigPin = 9; //pin trigger
const int echoPin = 10; //pin echo
long duration;
int distance;
void setup() {
 pinMode(trigPin, OUTPUT);
 pinMode(echoPin, INPUT);
 Serial.begin(9600);
}
void loop() {
 // Clears pin trigger
 digitalWrite(trigPin, LOW);
 delayMicroseconds(2);
 //set pin trigger selama 10 microsecond
 digitalWrite(trigPin, HIGH);
 delayMicroseconds(10);
 digitalWrite(trigPin, LOW);
 //membaca jarak yang dihasilkan
 duration = pulseIn(echoPin, HIGH);
 //memperkirakan jarak
 distance= (duration*0.034)/2;
 //tampilkan diserial monitor
 Serial.print("Distance: ");
 Serial.print(distance);
 Serial.println("cm");
}
```

lat_12_2

```
const int trigPin = 9; //pin trigger
const int echoPin = 10; //pin echo
int piezo=6; //Piezo speaker di pin 6.
long duration;
int distance;
void setup() {
 pinMode(trigPin, OUTPUT);
 pinMode(echoPin, INPUT);
 pinMode(piezo, OUTPUT);
 Serial.begin(9600);
}
void loop() {
 // Clears pin trigger
 digitalWrite(trigPin, LOW);
 delayMicroseconds(2);
 //set pin trigger selama 10 microsecond
 digitalWrite(trigPin, HIGH);
 delayMicroseconds(10);
 digitalWrite(trigPin, LOW);
 //membaca jarak yang dihasilkan
 duration = pulseIn(echoPin, HIGH);
 //memperkirakan jarak
 distance= (duration*0.034)/2;
 //tampilkan diserial monitor
 if (distance >= 200 || distance <= 0)
 {
 Serial.println("No measurement");
 }
 else
 {
```

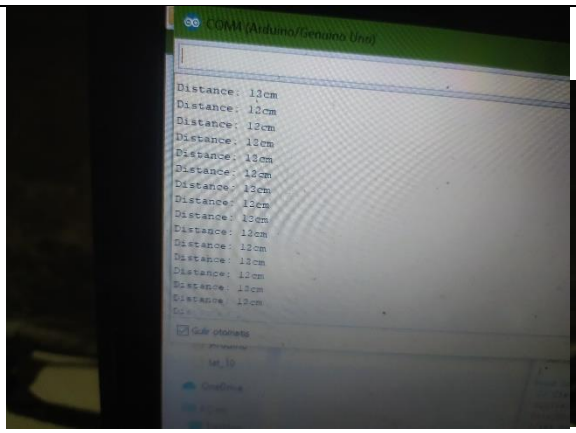
Bagian 1 Bagian 2

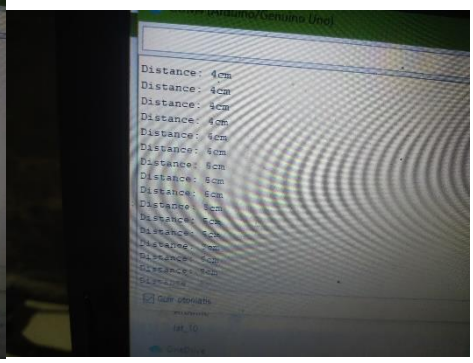
3. Hasil screenshot dokumentasi video praktikum saat kode program pada Arduino IDE berhasil di-upload ke board Arduino.

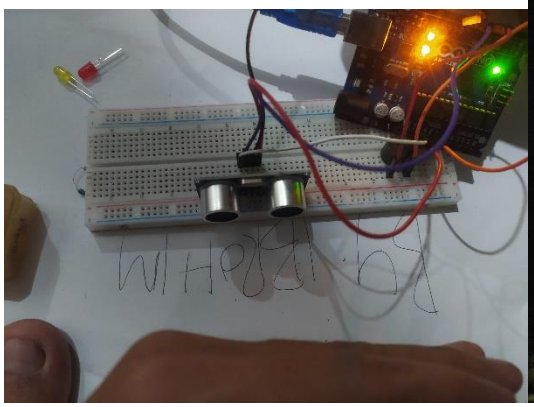


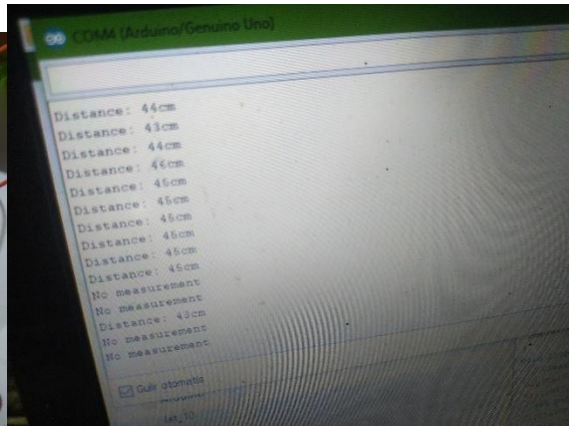
Bagian 1

Pada Rangkaian ini, ketika ojek benda di letakan tepat lurus dengan sensor jarak, maka pada serial monitor akan mencatat hasil pengukuran jarak objek dengan sensor tersebut.









Bagian 2

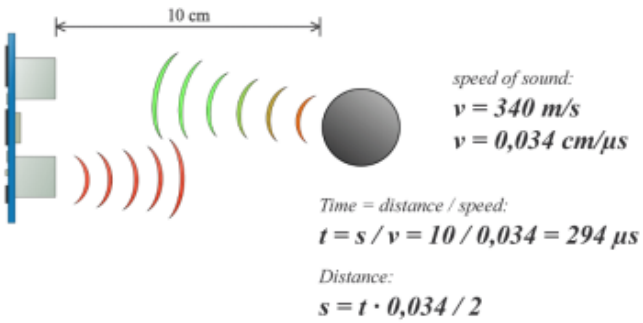
Sama seperti bagian 1, hanya saja di bagian ini ditambahkan buzzer, yang mana akan ada fungsi tambahan dimana ketika objek benda lebih dekat kurang dari 6 cm maka buzer akan berbunyi.

Analisis

Bagian 1

Cara kerja singkat dari sensor ini yaitu ketika pin trigger di beri sinyal pendek (5v) dari mikrokontroler maka sensor mengeluarkan gelombang suara, dan ketika gelombang tersebut memantul mengenai dinding atau objek lainya gelombang tersebut kembali, sensor akan mengirim sinyal balik ke arduino melalui pin echo.

Arduino mengukur waktu antara transmisi dan kembalinya gelombang suara dan mengonversi waktu tersebut menjadi jarak. Perhitungan untuk mendapatkan jarak dari sensor ultrasonik ini sebagai berikut.



Untuk penjelasan programnya, bagian awal (header program) dimuai dengan pendefinisian variabel const int trigPin = 9; yang berarti pin 9 dijadikan pin triger sensor ultrasonik. Lalu const int echoPin = 10; yang berato pin 10 dijadikan pin echo sensor ultrasonil. Kemudian long duration; yang fungsinya varibel penampung nilai durasi sinyal ultrasonik. Dam variabel int distance; yang berati variabel penampung nilai dari jaraknya.

```
const int trigPin = 9; //pin trigger
const int echoPin = 10; //pin echo
long duration;
int distance;
```

Pada bagian selanjutnya yakni Void setup di baris ke-5 sampai ke-9 ditulis sebuah pin sebagai input atau ouput, dengan menggunakan perintah pinmode, pada variabel trigPin di set sebagai sebagai output dan variabel echoPin sebagai input.

```
5 void setup() {
6  pinMode(trigPin, OUTPUT);
7  pinMode(echoPin, INPUT);
8  Serial.begin(9600);
9 }
```

Terakhir, pada bagian loop ditulis kode program yang akan dieksekusi terus menerus. Yang mana perintahnya dijelaskan pada gambar berikut:

```
10 void loop() {
11  // Clears pin trigger
12  digitalWrite(trigPin, LOW);
13  delayMicroseconds(2);
14  //set pin trigger selama 10 microsecond
15  digitalWrite(trigPin, HIGH);
16  delayMicroseconds(10);
17  digitalWrite(trigPin, LOW);
18  //membaca jarak yang dihasilkan
19  duration = pulseIn(echoPin, HIGH);
20  //memperkirakan jarak
21  distance= (duration*0.034)/2;
22  //tampilkan diserial monitor
23  Serial.print("Distance: ");
24  Serial.print(distance);
25  Serial.println("cm");
26 }
```

Dimana pada baris 11 sampai baris 21 merupakan perintah untuk membaca dan menghitung sensor kecepatan. Lalu perintah pada baris 22 samapai baris 26 adalah perintah untuk menampilkan nilai jarak pada LCD.

Bagian 2

Sama seperti skematik latihan sebelumnya, akan tetapi kita akan menambahkan buzzer, kaki positif buzzer dihubungkan ke pin digital 6 arduino, danpinnegatif ke ground. Perhatikan lambang positif pada buzzer menunjukan kaki positif pada buzzer.

Sama seperti pada bagian 1, tetapi terdapat tamabahan variabel *int piezo=6;* pada pin 6 yang merupakan variabel untuk menambahkan speaker, sepeti berikut:

```
1 const int trigPin = 9; //pin trigger
2 const int echoPin = 10; //pin echo
3 int piezo=6; //Piezo speaker di pin 6.
4 long duration;
5 int distance;
```

Lalu pada bagian void setup juga sama, dan terdapat tambahan perintah pada baris ke-9 dengan kode program *pinMode(piezo, OUTPUT);* yang fungsinya untuk mengeset variabel piezo sebagai output (buzzer).

```
6  void setup() {
7   pinMode(trigPin, OUTPUT);
8   pinMode(echoPin, INPUT);
9   pinMode(piezo, OUTPUT);
10  Serial.begin(9600);
11 }
```

Dan pada bagian void loop juga sama awalnya, lalu ditambhakan fungsi perualangan if yang tujuanya untuk mengatur buzzer jika ada objek benda pada jarak kurang dari 6c maka buzzer akan berbunyi. Dan selain itu disisipkan juga kode untuk menampilkan pengukuran pada LCD. kode tambahan ini ada pada baris ke 25 sampai baris ke 46.

```
19 digitalWrite(trigPin, LOW);
20 //membaca jarak yang dihasilkan
21 duration = pulseIn(echoPin, HIGH);
22 //memperkirakan jarak
23 distance= (duration*0.034)/2;
24 //tampilkan diserial monitor
25 if (distance >= 200 || distance <= 0)
26 {
27 Serial.println("No measurement");
28 }
29 else
30 {
31 Serial.print("Distance: ");
32 Serial.print(distance);
33 Serial.println("cm");
34 }
35 if (distance <= 6)
36 //Jika jarak kurang atau sama dengan 6cm
37 {
38 digitalWrite(piezo,HIGH);
39 //piezo akan berbunyi
40 }
41 else //jika tidak
42 {
43 digitalWrite(piezo,LOW);
44 //piezo berhenti berbunyi
45 }
46 }
```

Kesimpulan:

Berdasarkan praktikum ultrasonik ini memahami cara kerja dari sensor ultrasonik dengan arduino, sehingga dengan melakukan konfigurasi perhitungan nilai sensor ultrasonik kita dapat melakukan kalibrasi sensor dengan memakai rumus yang telah dibahas diatas. Selain itu juga dengan mengintegrasikan ke serial monitor kita dapat membuat alat ukur jarak. Dan dengan menambahkan buzzer serta menambahkan sedikit kode if pada program, kita juga bisa

membuat alat ukur jarak dengan suara buzzer jika terlalu dekat. Sehingga kedepannya dengan begi dapat diterapkan ke project arduino yang lebih kompleks.

Praktikum Sensor dan Sistem Mikroprosessor

Arduino Uno R3 Strater Kit

Latihan 13

Nama : Ibrohim
NIM : 1806068
Kelas : Elektronika Industri 2018

Judul : Ultrasonic Reverse Warning System

Tujuan:

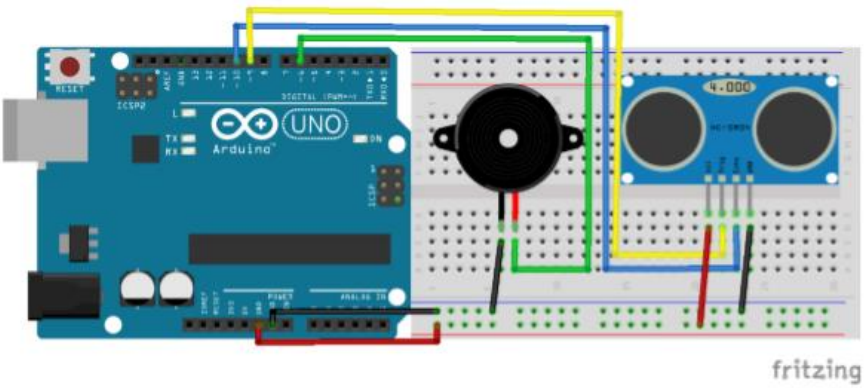
1. Menggunakan ultrasonik pada Arduino.
2. Membuat program ultrasonik dengan pengkondisian.
3. Membuat program Ultrasonic Reverse Warning System dengan menggunakan LED internal dan sensor ultrasonik.

Alat dan Bahan :

1. Breadboard 1 buah
2. Ultrasonik 1 buah
3. Buzzer 1 buah
4. Kabel jumper secukupnya

Langkah Kerja:

1. Siapkan alat dan bahan
2. Buatlah rangkaian seperti di bawah ini pada breadboard.



Sama seperti skematik latihan sebelumnya, akan tetapi kita akan menambahkan buzzer, kaki positif buzzer dihubungkan ke pin digital 6 arduino, danpinnegatif ke ground. Perhatikan lambang positif pada buzzer menunjukan kaki positif pada buzzer.

```
const int trigPin = 9; //pin trigger
const int echoPin = 10; //pin echo
int piezo=6; //Piezo speaker di pin 6.
long duration;
int distance;

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(piezo, OUTPUT);
  Serial.begin(9600);
}
```

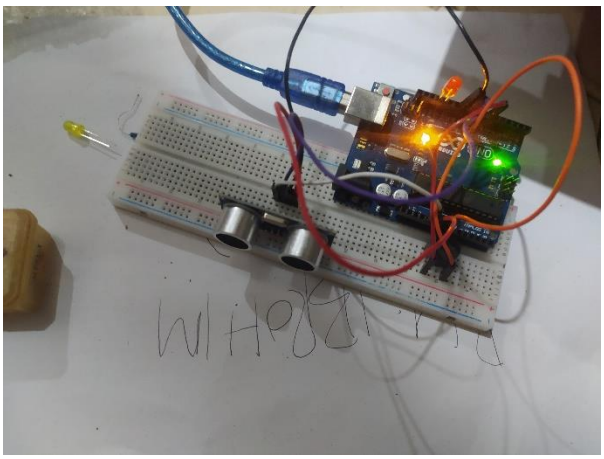
```
void loop() {
  // Clears pin trigger
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  //set pin trigger selama 10 microsecor
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  //membaca jarak yang dihasilkan
  duration = pulseIn(echoPin, HIGH);
  //memperkirakan jarak
  distance= (duration*0.034)/2;
  //tampilkan diserial monitor
```

```
if (distance >= 200 || distance <= 0)
{
  Serial.println("No measurement");
}
else
{
Serial.print("Distance: ");
Serial.print(distance);
  Serial.println("cm");
  }
if (distance <= 6)
//Jika jarak kurang atau sama dengan 6cm
{
  digitalWrite(piezo,HIGH);
  //piezo akan berbunyi
}
else //jika tidak
{
  digitalWrite(piezo,LOW);
  //piezo berhenti berbunyi
}
}
```

3. Compile dan upload kode program tersebut satu persatu (4 project) ke board Arduino Uno.
4. Amati apa yang terjadi pada praktikum ini dan buatlah kesimpulan.

Hasil Praktikum:

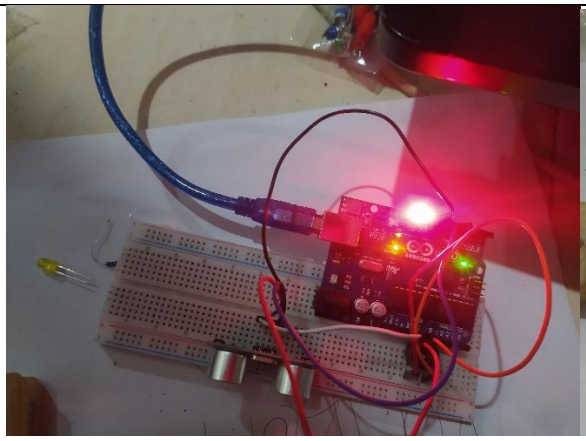
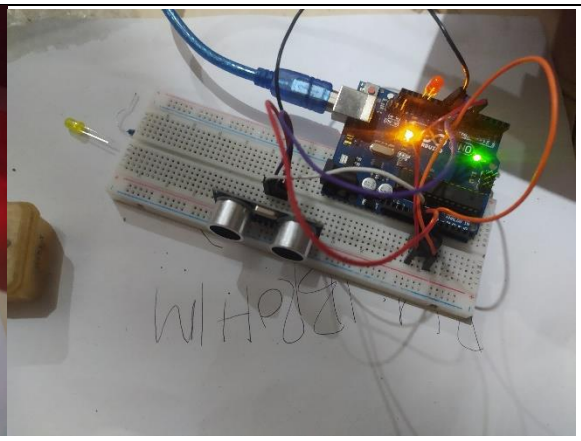
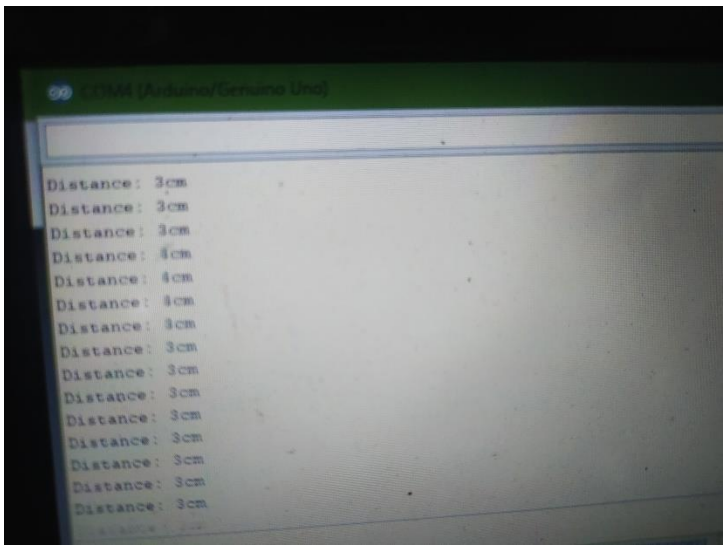
1. Hasil rangkaian pada breadboard.



2. Hasil kode program yang sudah berhasil di-compile.

```
const int trigPin = 9; //pin trigger
const int echoPin = 10; //pin echo
int piezo=6; //Piezo speaker di pin 5
int led=13; //Pin LED
long duration;
int distance;
void setup() {
 pinMode(trigPin, OUTPUT);
 pinMode(echoPin, INPUT);
 pinMode(piezo, OUTPUT);
 pinMode(led, OUTPUT);
 Serial.begin(9600);

void loop() {
 digitalWrite(trigPin, LOW); //Clears pin trigger
 delayMicroseconds(2);
 //set pin trigger selama 10 microsecond
 digitalWrite(trigPin, HIGH);
 delayMicroseconds(10);
 digitalWrite(trigPin, LOW);
 //membaca jarak yang dihasilkan
 duration = pulseIn(echoPin, HIGH);
 distance= (duration*0.034)/2; //memperkirakan jarak
 if (distance >= 200 || distance <= 0){
 Serial.println("No measurement");
 }
 else{
 Serial.print("Distance: ");
Serial.print(distance);
Serial.println("No measurement");
}
else{
Serial.print("Distance: ");
Serial.print(distance);
Serial.println("cm");
}
if (distance <= 20){
//Jika jarak kurang atau sama dengan 20cm
//piezo dan led akan hidup
digitalWrite(piezo,HIGH);
digitalWrite(led, HIGH);
delay(distance*10);
digitalWrite(piezo,LOW);
digitalWrite(led, LOW);
delay(distance*10);
}
else {//jika tidak
digitalWrite(piezo,LOW);
digitalWrite(led, LOW);
}
}
```

3. Hasil screenshot dokumentasi video praktikum saat kode program pada Arduino IDE berhasil di-upload ke board Arduino.

Pada program ini, efeknya hanya bisa dilihat pada video dokementasi dimana intinya adalah ketika kita mendekatkan objek ke sensor ultrasonik, maka kedipan led internel arduino dan kedipan nyala buzzer akan semakin cepat, semakin dekat objeknya semakin cepat kedipannya.

Analisis

Latihan kali ini masih sama seperti latihan 12 yang menggunakan ultrasonik juga. Hanya ada sedikit tambahan led dan pada pengkondisiannya. Dimana cara kerjanya yaitu ketika posisi jarak ultrasonik sudah pada jarak tertentu maka led akan berkedip, jika makin dekat kedipan led akan semakin cepat, begitu sebaliknya.

Untuk penjelasan programnya, bagian awal (header program) dimuai dengan pendefinisian variabel const int trigPin = 9; yang berarti pin 9 dijadikan pin triger sensor ultrasonik. Lalu const int echoPin = 10; yang berato pin 10 dijadikan pin echo sensor ultrasonik. Lalu Variabel *int piezo=6;* Untuk Piezo speaker di pin 5. Dan *int led=13;* untuk set Pin LED. Kemudian long duration; yang fungsinya varibel penampung nilai durasi sinyal ultrasonik. Dam variabel int distance; yang berati variabel penampung nilai dari jaraknya.

```
const int trigPin = 9; //pin trigger
const int echoPin = 10; //pin echo
int piezo=6; //Piezo speaker di pin 5
int led=13; //Pin LED
long duration;
int distance;
```

Pada bagian selanjutnya yakni Void setup di baris ke-5 sampai ke-9 ditulis sebuah pin sebagai input atau ouput, dengan menggunakan perintah pinmode, pada variabel trigPin di set sebagai sebagai output dan variabel echoPin sebagai input, variabel piezo di set sebagai output, variabel led di set sebagai output. Dan memanggil serial monitor *Serial.begin(9600);*.

```
7  void setup() {
8   pinMode(trigPin, OUTPUT);
9   pinMode(echoPin, INPUT);
10  pinMode(piezo, OUTPUT);
11  pinMode(led, OUTPUT);
12  Serial.begin(9600);
13  }
```

Terakhir, pada bagian loop ditulis kode program yang akan dieksekusi terus menerus. Yang mana perintahnya dijelaskan pada gambar berikut:

```
14 void loop() {
15   digitalWrite(trigPin, LOW); //Clears pin trigger
16   delayMicroseconds(2);
17   //set pin trigger selama 10 microsecond
18   digitalWrite(trigPin, HIGH);
19   delayMicroseconds(10);
20   digitalWrite(trigPin, LOW);
21   //membaca jarak yang dihasilkan
22   duration = pulseIn(echoPin, HIGH);
23   distance= (duration*0.034)/2; //memperkirakan jarak
24   if (distance >= 200 || distance <= 0){
25   Serial.println("No measurement");
26   }
27   else{
28   Serial.print("Distance: ");
29  Serial.print(distance);
30  Serial.println("cm");
31   }
32   if (distance <= 20){
33   //Jika jarak kurang atau sama dengan 20cm
34   //piezo dan led akan hidup
35   digitalWrite(piezo,HIGH);
36   digitalWrite(led, HIGH);
37   delay(distance*10);
38   digitalWrite(piezo,LOW);
39   digitalWrite(led, LOW);
40   delay(distance*10);
41   }
42   else {//jika tidak
43   digitalWrite(piezo,LOW);
44   digitalWrite(led, LOW);
45   }
46  }
```

Kesimpulan:

Berdasarkan praktikum tesebut, kita dapat mengembangkan program Ultrasonic Reverse Warning System dengan menggunakan LED internal dan sensor ultrasonik. Dimana dengan memahami contoh tersebut tentu kita dapat menerapkanya ke berbagai ultrasonik project arduino yang lebih kompleks.

Praktikum Sensor dan Sistem Mikroprosessor

Arduino Uno R3 Strater Kit

Latihan 14

Nama : Ibrohim
NIM : 1806068
Kelas : Elektronika Industri 2018

Judul : Relay

Tujuan:

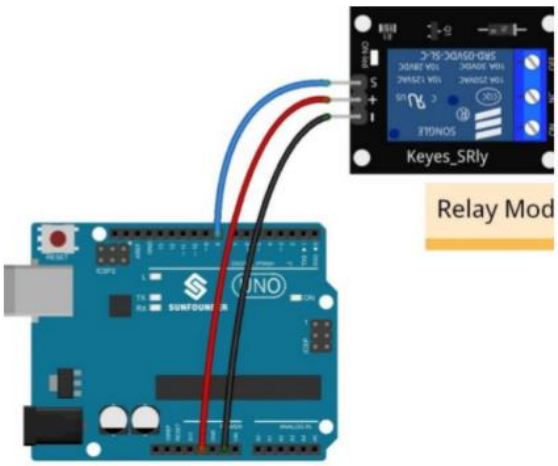
1. Menggunakan Arduino sebagai relay.
2. Membuat program sederhana untuk menyalakan dan mematikan relay.

Alat dan Bahan :

1. Breadboard 1 buah
2. Relay 1 buah
3. Kabel jumper secukupnya

Langkah Kerja:

1. Siapkan alat dan bahan
2. Buatlah rangkaian seperti di bawah ini pada breadboard.



Hubungkan input vcc dan ground di modul relay ke arduino, lalu hubungkan pin IN ke pin 8 Arduino, pada modul relay ini mempunyai 3 pin input yaitu vcc, ground dan sinyal, sinyal ini diberi trigger oleh mikrokontroler 5V atau 0V.

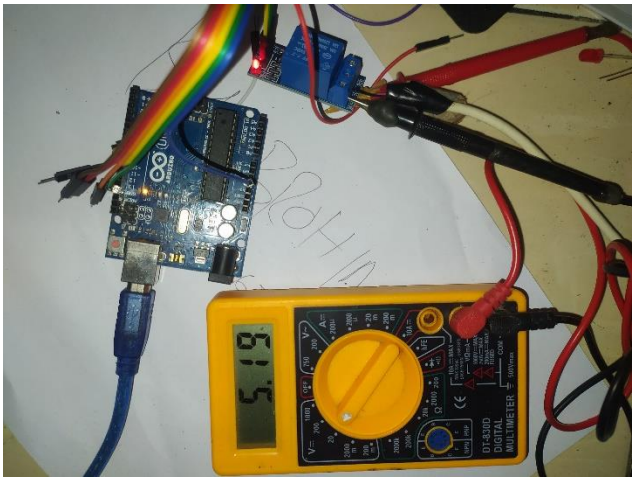
3. Masukan kode program di bawah ini ke Arduino IDE.

```
void setup()
{
  pinMode(8, OUTPUT);
}
void loop()
{
  digitalWrite(8, HIGH); //relay hidup
  delay(1000);
  digitalWrite(8, LOW); //matikan relay
  delay(1000);
}
```

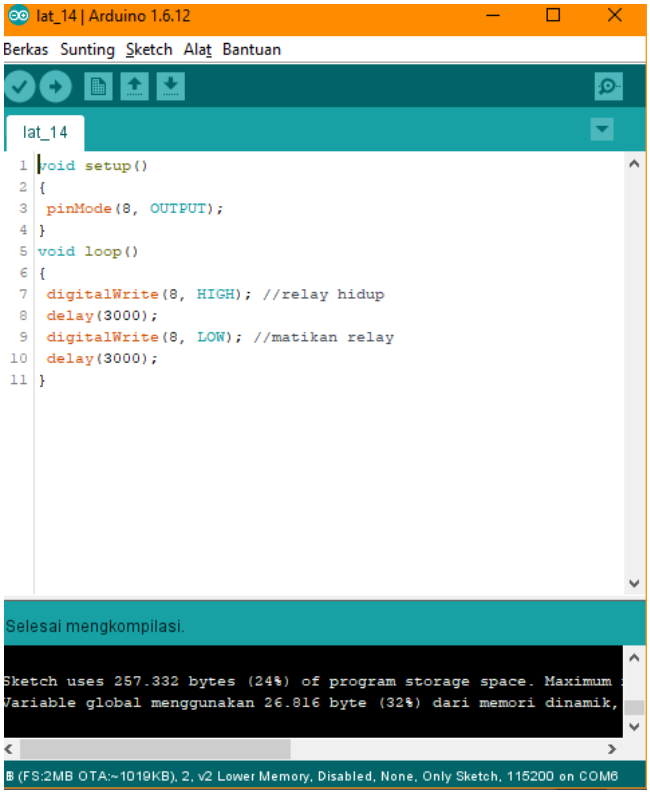
4. Compile dan upload kode program tersebut ke board Arduino Uno.
5. Amati apa yang terjadi pada praktikum ini dan buatlah kesimpulan.

Hasil Praktikum:

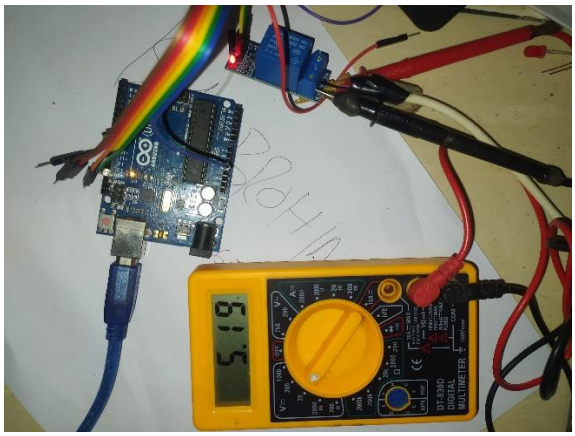
1. Hasil rangkaian pada breadboard.



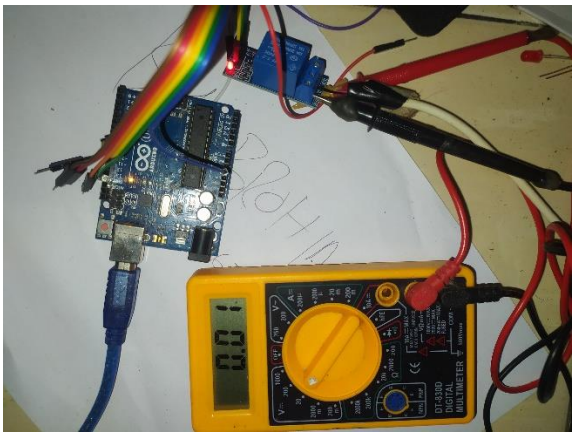
2. Hasil kode program yang sudah berhasil di-compile.



3. Hasil screenshot dokumentasi video praktikum saat kode program pada Arduino IDE berhasil di-upload ke board Arduino.



Relay menyala selama satu detik (NC)



Lalu Relay Mati selama 1 detik (NO).

Analisis

Relay adalah salah satu aktuator yang banyak digunakan, fungsi relay digunakan untuk mengsaklar, yang bisa di saklar tidak hanya tegangan DC tapi juga tegangan AC dengan trigger yang kecil, penggunaanya hanya memberi logika High ata Low pada pin input relay.

Relay mempunyai 2 kondisi Normally Open (NO) dan Normally Close (NC), maksudnya adalah ketika posisi relay tidak mendapat tegangan input kondisi relay saat itu adalah tidak terhubung atau Normally open, sedangkan Normally Close sebaliknya ketika relay tidak mendapat tegangan input maka kondisi relay sudah terhubung.

Pada rangkaian sederhana ini terdapat kode program yang terbagi menjadi 3 blok, yaitu Header, Setup, dan Loop. Pada bagian header program ini tidak diisi karena memamng pada relay nilainya adalah bolean (true atau false) sehingga pendeklarasian variabel pada program sederhana tidak perlu. Pada bagian setup ditulis sebuah pin sebagai ouput dengan menggunakan perintah *pinMode* pada pin 8. Terakhir, pada bagian loop ditulis kode program yang akan dieksekusi terus menerus, dimana pada program ini terdapat perintah *digitalWrite(8, HIGH);* untuk menghidupkan relay (NC), lalu program *delay(3000);* untuk mendelay 3 detik hidup, kemudian programm *digitalWrite(8, LOW);* untuk mematikan relay (NO) dilanjutkan dengan *delay(3000);* untuk mendelay 3 detik mati. dan berulang seterusnya.

Kesimpulan:

Berdasarkan praktikunm ini, dapat disimpulkan bahwa dengan adanya relay, kita dapat membuat saklar otomatis hanya dengan menggunakan kode program yang sederhana. Selain itu juga kita memahami bagaima cara kerja saklar, dan penulisan programnya pada arduino. Dimana dengan memahami contoh tersebut tentu kita dapat menerapkanya ke berbagai ultrasonik project arduino yang lebih kompleks.

Praktikum Sensor dan Sistem Mikroprosessor

Arduino Uno R3 Strater Kit

Latihan 15

Nama : Ibrohim
NIM : 1806068
Kelas : Elektronika Industri 2018

Judul : DHT Temperature and Humadity

Tujuan:

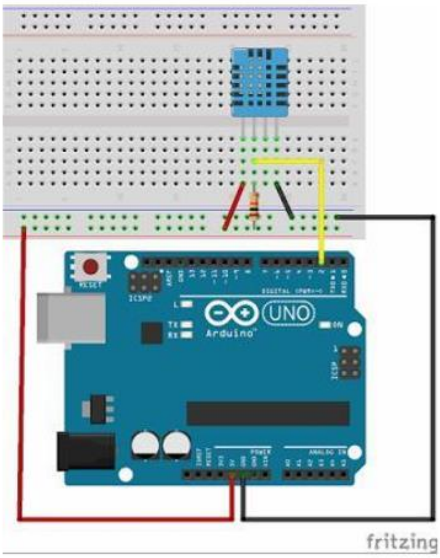
1. Menggunakan Arduino sebagai sensor suhu dan kelembaban.
2. Membuat program sederhana untuk DHT Temperature and Humadity

Alat dan Bahan :

1. Breadboard 1 buah
2. Sensor DHT11 1 buah
3. Kabel jumper secukupnya

Langkah Kerja:

1. Siapkan alat dan bahan
2. Buatlah rangkaian seperti di bawah ini pada breadboard.



- Hubungkan kaki ke-1 DHT11 ke 5V Arduino
- Hubungkan kaki ke-2 DHT11 ke pin 2 Arduino Breadboard DHT11 Kabel Jumper 83 Dengan menggunakan Resisitor 10 Kilo Ohm, hubungkan kaki ke-2 DHT11 ke 5V Arduino.

- Hubungkan kaki ke-4 DHT11 ke GND Arduino, Untuk kaki Ke-3 DHT11 kita abaikan sajaProgram.

3. Masukan kode program di bawah ini ke Arduino IDE.

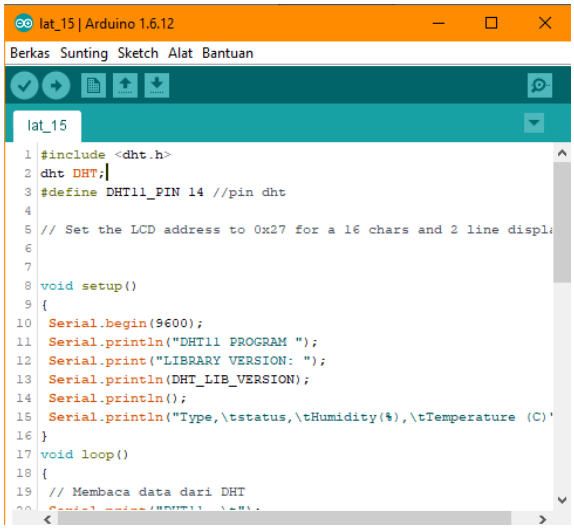
```
void setup()
{
  pinMode(8, OUTPUT);
}
void loop()
{
  digitalWrite(8, HIGH); //relay hidup
  delay(1000);
  digitalWrite(8, LOW); //matikan relay
  delay(1000);
}
```

4. Compile dan upload kode program tersebut ke board Arduino Uno.
5. Amati apa yang terjadi pada praktikum ini dan buatlah kesimpulan.

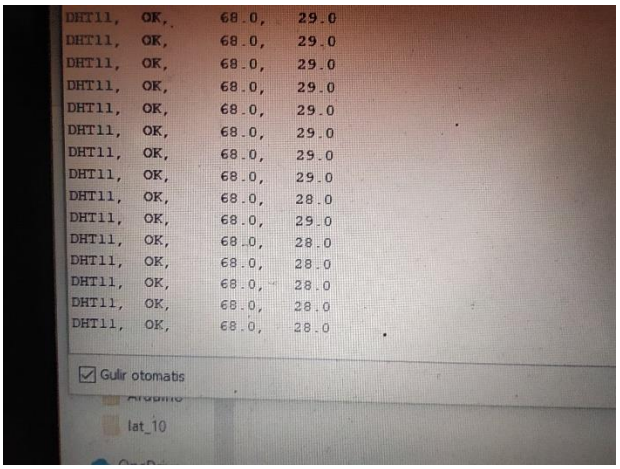
Hasil Praktikum:

1. Hasil rangkaian pada breadboard.

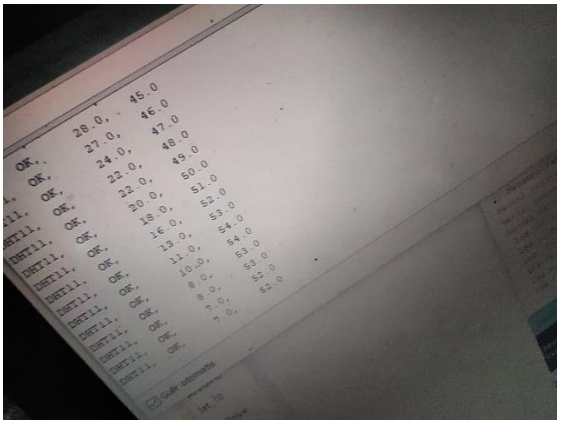
2. Hasil kode program yang sudah berhasil di-compile.



3. Hasil screenshot dokumentasi video praktikum saat kode program pada Arduino IDE berhasil di-upload ke board Arduino.



Kelembaban yang terukur pada suhu ruangan (suhu kamar saya



Setelah DHT11 dipanaskan dengan blower (solder uap), beberapa detik kemudian suhunya berubah seperti pada gambar.

Analisis

Pada latihan kali ini kita akan membuat sebuah pengukur suhu dan kelembaban menggunakan sensor DHT11.

Pada rangkaian sederhana ini terdapat kode program yang terbagi menjadi 3 blok, yaitu Header, Setup, dan Loop. Pada bagian header program terdapat berikut:

```
lat_15
1 #include <dht.h>
2 dht DHT;
3 #define DHT11_PIN 14 //pin dht
4
5 // Set the LCD address to 0x27 for
```

Perintah tersebut merupakan perintah untuk mengimpot dan menyeting library dari sensor DHT11 pada arduino.

Pada bagian setup ditulis sebuah program untuk mengaktifkan serial monitor (baris 10) dan program untuk menuliskan setingan DHT11 pada serial tesrbut dengan perintah *println*.(baris 11-15).

```
8  void setup()
9  {
10  Serial.begin(9600);
11  Serial.println("DHT11 PROGRAM ");
12  Serial.print("LIBRARY VERSION: ");
13  Serial.println(DHT_LIB_VERSION);
14  Serial.println();
15  Serial.println("Type,\tstatus,\tHumidity(%),\tTemperature (
16 }
```

Terakhir, pada bagian loop ditulis kode program yang akan dieksekusi terus menerus, dimana pada program ini terdapat perintah untuk membaca data pada pin sensor dht 11 dengan perintah berikut:

```
17 void loop()
18 {
19  // Membaca data dari DHT
20  Serial.print("DHT11, \t");
21  int chk = DHT.read11(DHT11_PIN);
```

Setelah itu pada bagian selanjutnya akan mengecek data pada sensor dht11 apakah terdapat nilai yang eror, dengan menggunakan perintah pengkondisian case berikut:

```
22 //cek data sensor DHT
23 switch (chk)
24 {
25 case DHTLIB_OK:
26 Serial.print("OK,\t");
27 break;
28 case DHTLIB_ERROR_CHECKSUM:
29 Serial.print("Checksum error,\t");
30 break;
31 case DHTLIB_ERROR_TIMEOUT:
32 Serial.print("Time out error,\t");
33 break;
34 case DHTLIB_ERROR_CONNECT:
35 Serial.print("Connect error,\t");
36 break;
37 case DHTLIB_ERROR_ACK_L:
38 Serial.print("Ack Low error,\t");
39 break;
40 case DHTLIB_ERROR_ACK_H:
41 Serial.print("Ack High error,\t");
42 break;
43 default:
44 Serial.print("Unknown error,\t");
45 break;
46 }
```

Setelah data dicek dengan benar, data tersebut kemudian akan ditampilkan pada serial monitor atasu LCD, seperti pada perintah berikut:

```
// Menampilkan data dari sensor DHT
Serial.print(DHT.humidity, 1);
Serial.print(",\t");
Serial.println(DHT.temperature, 1);

delay(2000);
}
```

Kesimpulan:

Berdasarkan praktikunm ini, dapat disimpulkan bahwa dengan menggunakan sensor DHT11 kita bisa membuat alat untuk mengukur suhu dan kelembaban di sekitar secara real time. Selain itu juga dengan melakukan praktikum ini kita bisa membuat program dari sensor DHT11 pada arduino yang pada pembuatannya terdapat setingan – setingan yang cukup banyak. Dengan begitu dengan melakukan praktikum ini kita dapat memahami contoh tersebut tentu kita dapat menerapkanya ke berbagai ultrasonik project arduino yang lebih kompleks.

Praktikum Sensor dan Sistem Mikroprosessor

Arduino Uno R3 Strater Kit

Latihan 16

Nama : Ibrohim
NIM : 1806068
Kelas : Elektronika Industri 2018

Judul : Infrared

Tujuan:

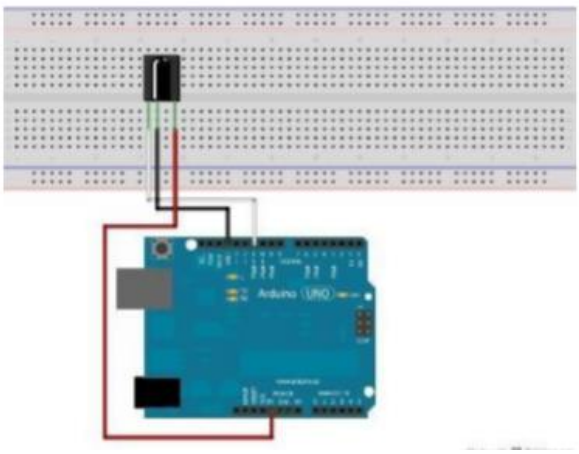
1. Menggunakan Arduino sebagai sensor infrared.
2. Membuat program sederhana untuk membuat pemancar penerima Infrared.

Alat dan Bahan :

1. Breadboard 1 buah
2. Sensor Infrared 1 buah
3. Kabel jumper secukupnya

Langkah Kerja:

1. Siapkan alat dan bahan
2. Buatlah rangkaian seperti di bawah ini pada breadboard.



   Hubungkan pin 1 sensor ke pin 11 arduino, pin 2 ke ground serta 3 ke vcc.

3. Masukan kode program di bawah ini ke Arduino IDE.

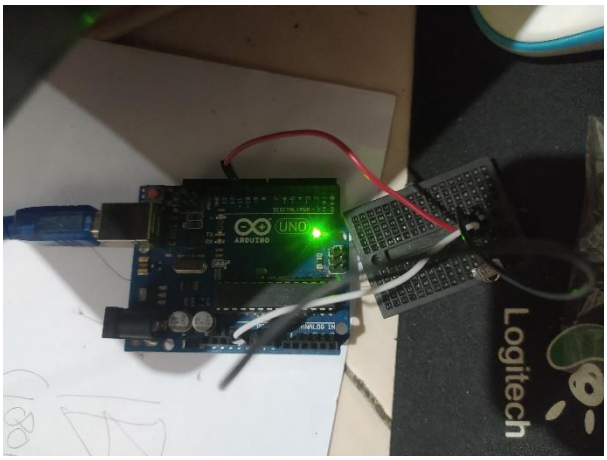
```
//Infrared Receiver
#include <Irremote.h>
int RECV_PIN = 11; //pin sensor
IRrecv irrecv(RECV_PIN);
decode_results results;

void setup ()
{
  Serial.begin(9600);
  pinMode (13, OUTPUT);
  irrecv.enableIRIn();
}

void loop()
{
  if (irrecv.decode(&results)) {
    Serial.println(results.value, DEC);
    irrecv.resume();
  }
}
```

4. Compile dan upload kode program tersebut ke board Arduino Uno.
5. Setelah berhasil upload, silahkan buka serial monitor arduino lalu tekat sembarang tombol pada remote infrared dan arahkan ke sensor TL1838, maka akan tampil kode berbentuk angka desimal, setiap tombol yang ditekan akan menghasilkan angka desimal yang berbeda, angka itulah yang akan kita definisikan nanti.

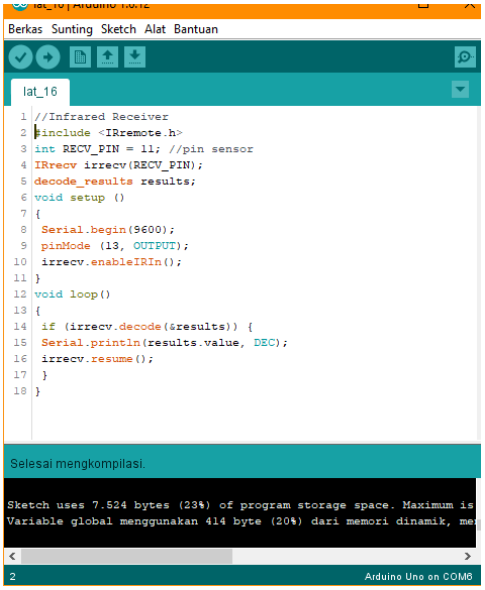
   Contoh ketika menekan tombol “1” maka muncul nilai desimal “1678996467”, Setiap remote infrared mempunyai nilai desimal yang berbeda, terkadang ada juga ketika tombol ditekan menghasilkan nilai yang tidak konstan atau bahkan muncul nilai “4294967295” ini bukan kode unik dari tombol tersebut, hal ini bisa terjadi biasanya apabila kita menekan tombol secara berkelanjutan/ditekan lama. Dengan kata lain sama seperti type sensor lain, remote pun punya spesifikasinya masing masing.
6. Amati apa yang terjadi pada praktikum ini dan buatlah kesimpulan.

Hasil Praktikum:

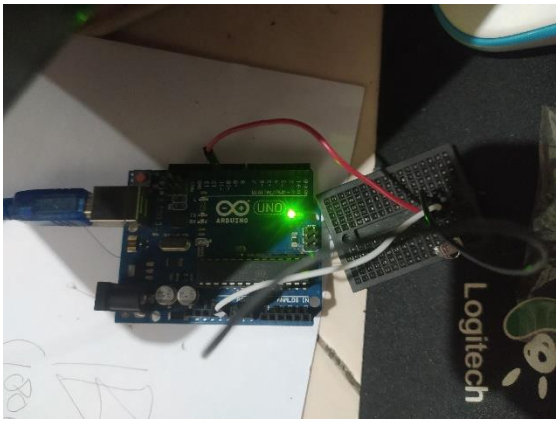
1. Hasil rangkaian pada breadboard.

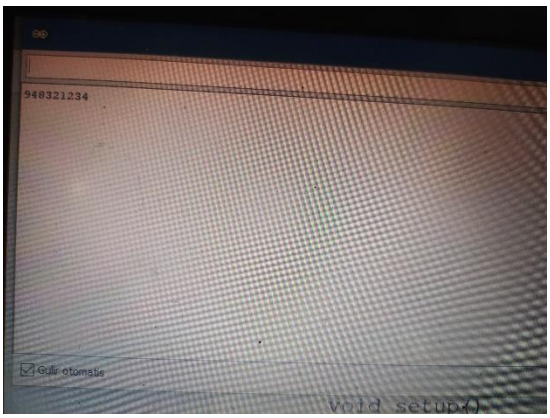


2. Hasil kode program yang sudah berhasil di-compile.

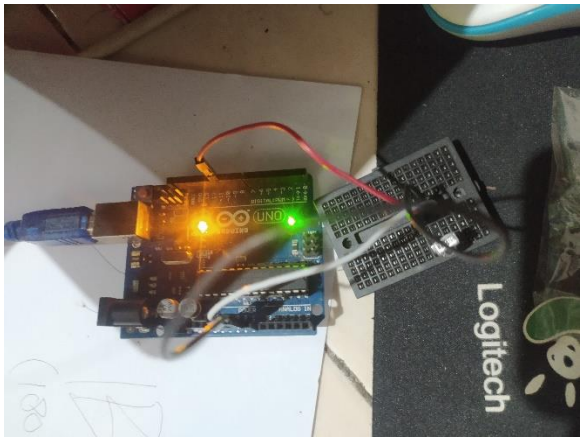


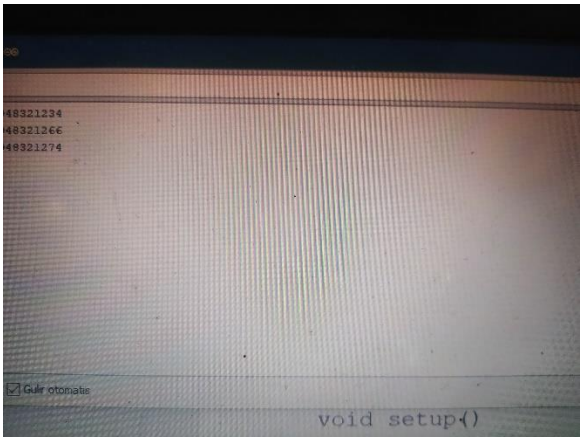
3. Hasil screenshot dokumentasi video praktikum saat kode program pada Arduino IDE berhasil di-upload ke board Arduino.





Pada gambar tersebut, ketika mengarahkan remot TV changhong ke sensor infrared, lalu menekan tombol power pada remot maka serial monitor akan membaca kode dari tobol remot tersebut yakni *948321234*





Kemudian lanjut dengan menekan tombol lain, yakni tobol ganti chanel atas pada remot, sehingga pada serial monitor membaca kode 48321266

Analisis

Pada latihan kali ini kita akan menggunakan sensor TL1838 sebagai penerima sinyal infrared, sensor ini memiliki 3 pin, konfigurasi dari setiap pin ini yaitu 1=out/signal, 2=ground dan 3=vcc (lihat datasheet). Hati-hati untuk sensor type lain susunan pin bisa berubah, posisi ground dan vcc bisa tertukar, jadi lihat datasheet untuk memastikan konfigurasi untuk setiap sensor.

Pada rangkaian sederhana ini terdapat kode program yang terbagi menjadi 3 blok, yaitu Header, Setup, dan Loop. Pada bagian header program terdapat berikut:

```
1 //Infrared Receiver
2 #include <IRremote.h>
3 int RECV_PIN = 11; //pin sensor
4 IRrecv irrecv(RECV_PIN);
5 decode_results results;
```

Perintah tersebut merupakan perintah untuk mengimpot dan menyeting library dari sensor infrared receiver pada arduino.

Pada bagian void setup ditulis sebuah program untuk mengaktifkan serial monitor (baris 8) lalu perintah untuk mngeset pin 13 sebagai output sinyal yang ditangkap infrared dan perintah untuk menuliskan setingan DHT11 pada serial tesrbut dengan perintah pada baris 10.

```
5 decode_results results;
6 void setup ()
7 {
8  Serial.begin(9600);
9  pinMode (13, OUTPUT);
10  irrecv.enableIRIn();
11 }
```

Terakhir, pada bagian void loop ditulis kode program yang akan dieksekusi terus menerus, merupakan perintah untuk membaca sinyal infrared dan menuliskannya ke arduino, perintahnya sebagai berikut:

```
12 void loop()
13 {
14   //melakukan pengkondisian apakah ada sinyal infrared yang masuk
15   if (irrecv.decode(&results)) {
16     //mmembaca hasul dan menuliskannya(biner ke desimal) ke serial monitor
17     Serial.println(results.value, DEC);
18     irrecv.resume();
19   }
20 }
```

Kesimpulan:

Berdasarkan praktikunm ini, dapat disimpulkan bahwa dengan menggunakan sensor infrared kita bisa membuat penerima infrared menggunakan arduino. Selain itu juga dengan melakukan praktikum ini kita bisa melakukan setingan – setingan yang cukup banyak. Dengan begitu dengan melakukan praktikum ini kita dapat memahami contoh tersebut tentu kita dapat menerapkanya ke berbagai ultrasonik project arduino yang lebih kompleks.

Praktikum Sensor dan Sistem Mikroprosessor

Arduino Uno R3 Strater Kit

Latihan 17

Nama : Ibrohim
NIM : 1806068
Kelas : Elektronika Industri 2018

Judul : Control LED Using Infrared

Tujuan:

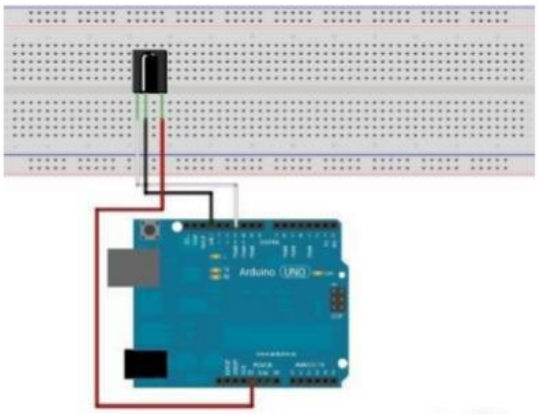
1. Menggunakan Arduino sebagai sensor Infrared.
2. Membuat program sederhana untuk menyalakan LED dengan Infrered receiver sensor.

Alat dan Bahan :

1. Breadboard 1 buah
2. Sensor DHT11 1 buah
3. LED Merah 1 buah
4. Kabel jumper secukupnya

Langkah Kerja:

1. Siapkan alat dan bahan
2. Buatlah rangkaian seperti di bawah ini pada breadboard.



Untuk skematik masih menggunakan skematik yang sama, kita juga bisa menggunakan LED internal pada arduino, yaitu pada pin digital 13 atau kita Breadboard TL1838 Kabel Jumper Remote Infrared (Remote Universal, Remot TV, dll) 88 bisa memasang LED lagi di pin yang lain seperti yang ada pada latihan sebelumnya.

3. Masukan kode program di bawah ini ke Arduino IDE.

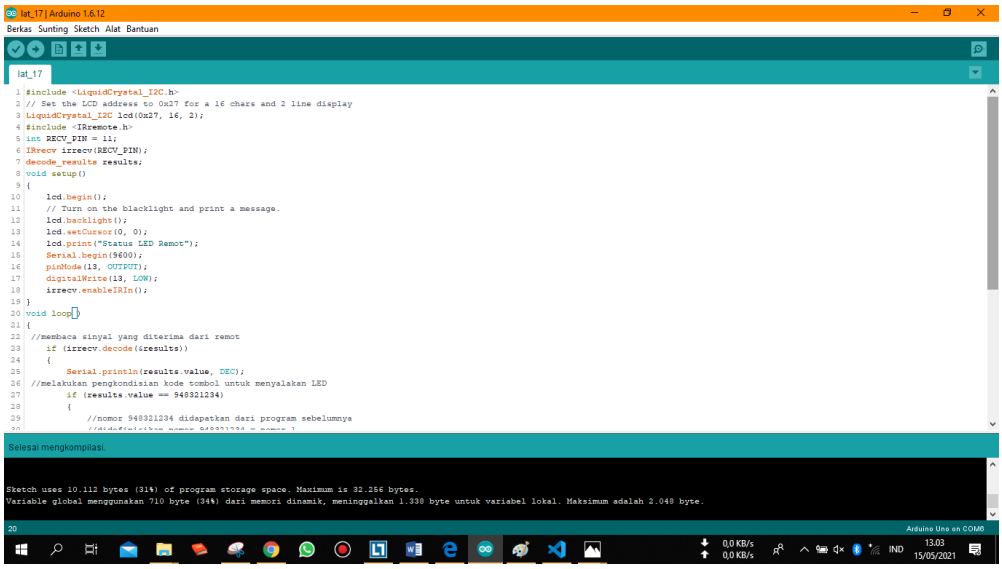
```
#include <Irremote.h>
int RECV_PIN = 11;
IRrecv irrecv(RECV_PIN);
decode_results results;

void setup()
{
  Serial.begin(9600);
  pinMode (13, OUTPUT);
  digitalWrite(13, LOW);
  irrecv.enableIRIn();
}
void loop() {
  if (irrecv.decode(&results)) {
    Serial.println(results.value, DEC);
  if (results.value == 16724175){
    //nomor 16724175 didapatkan dari program
sebelumnya
    //didefinisikan nomor 16724175 = nomor 1
    digitalWrite (13, HIGH);
  }//...led akan menyala

  if (results.value == 16718055)
    //nomor 16718055 didapatkan dari program
sebelumnya
    //didefinisikan nomor 16718055 = nomor 1.
    //jika receiver menerima kode ini maka
    {digitalWrite (13, LOW);}
    //….led akan mati
irrecv.resume(); // Receive the next value
}
```

4. Compile dan upload kode program tersebut ke board Arduino Uno.
5. Amati apa yang terjadi pada praktikum ini dan buatlah kesimpulan.

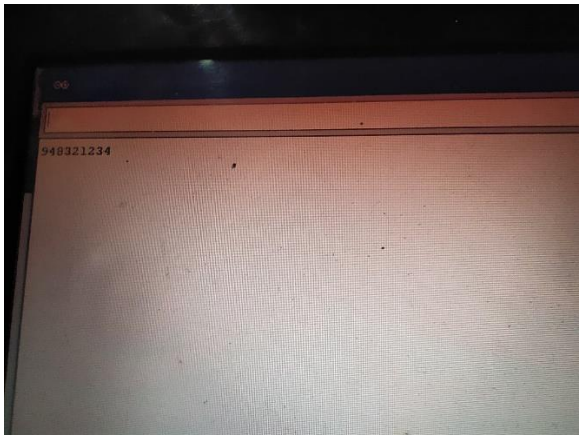
Hasil Praktikum:

1. Hasil rangkaian pada breadboard.

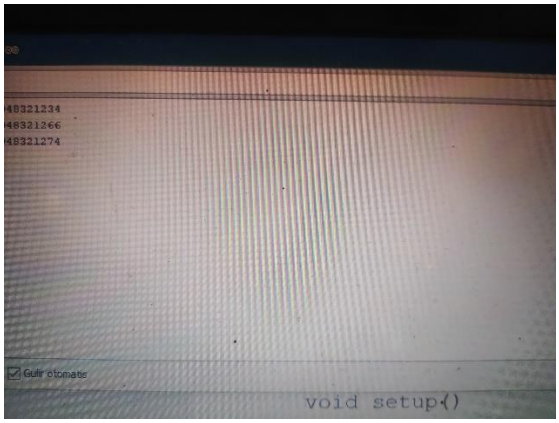
2. Hasil kode program yang sudah berhasil di-compile.



3. Hasil screenshot dokumentasi video praktikum saat kode program pada Arduino IDE berhasil di-upload ke board Arduino.



Pada gambar tersebut, pada saat saya menekan tombol power remot TV changhong maka sensor infrared akan menrima sinyal dari tombol tersebut yakni berupa kode 948321234, dimana kode tersebut diseting untuk menghidupkan LED pada arduino, sehingga LED menyala dan status pada LCD juga menyala.



Lalu pada selanjutnya gambar tersebut, pada saat saya menekan tombol volume atas remot TV changhong maka sensor infrared akan menrima sinyal dari tombol tersebut yakni berupa kode 948321274, dimana kode tersebut diseting untuk mematikan LED pada arduino, sehingga LED mati dan status pada LCD juga mati.

Analisis

Pada latihan ini melanjutkan latihan sebelumnya (latihan 15) dimana di latihan ini kita akan menggunakan remote infrared untuk mengendalikan LED.

Pada rangkaian ini terdapat kode program yang terbagi menjadi 3 blok, yaitu Header, Setup, dan Loop. Pada bagian header program terdapat berikut:

```
#include <LiquidCrystal_I2C.h>
// Set the LCD address to 0x27 for a 16 chars and 2 line display
LiquidCrystal_I2C lcd(0x27, 16, 2);
#include <IRremote.h>
int RECV_PIN = 11;
IRrecv irrecv(RECV_PIN);
decode_results results;
```

Perintah tersebut merupakan perintah untuk mengimpot dan menyeting library dari sensor infrared, LCD, dan menseting pin – pinya dengan perintah seperti pada gambar.

Pada bagian void setup ditulis sebuah program untuk mengaktifkan LCD dan serial monitor (baris 10 -15) dan perintah untuk menset pin 13 sebagai output untuk LEDnya, lalu perintah untuk mengeset pin 13 bernilai awal low. dan untuk menuliskan setingan infrared tesrbut dengan perintah pada baris ke 18.

```
 8 void setup()
 9 {
10     lcd.begin();
11     // Turn on the blacklight and print a message.
12     lcd.backlight();
13     lcd.setCursor(0, 0);
14     lcd.print("Status LED Remot");
15     Serial.begin(9600);
16     pinMode(13, OUTPUT);
17     digitalWrite(13, LOW);
18     irrecv.enableIRIn();
19 }
```

Terakhir, pada bagian void loop ditulis kode program yang akan dieksekusi terus menerus, dimana pada program ini terdapat perintah pengkondisian untuk menyalakan LED dari remot TV lalu mengirimkan status LEDnya ke LCD dan serial monitor, untuk penjelasanya lebih jelas seperti pada komen gambar berikut:

```
20 void loop()
21 {
22  //membaca sinyal yang diterima dari remot
23     if (irrecv.decode(&results))
24     {
25         Serial.println(results.value, DEC);
26  //melakukan pengkondisian kode tombol untuk menyalakan LED
27         if (results.value == 948321234)
28         {
29             //nomor 948321234 didapatkan dari program sebelumnya
30             //didefinisikan nomor 948321234 = nomor 1
31             digitalWrite(13, HIGH);
32             lcd.setCursor(0, 1);
33             lcd.print("LED Nyala Boss ");
34         } //...led akan menyala
35 //melakukan pengkondisian kode tombol untuk mamatikan LED
36         if (results.value == 948321266) {
37
38             //nomor 948321266 didapatkan dari program sebelumnya
39             //didefinisikan nomor 948321266 = nomor 1.
40             //jika receiver menerima kode ini maka
41             digitalWrite(13, LOW);
42             lcd.setCursor(0, 1);
43             lcd.print("LED Mati Boss ");
44             //....led akan mati
45         }
46 //membaca program selanjutnya
47             irrecv.resume(); // Receive the next value
48     }
49 }
```

Kesimpulan:

Berdasarkan praktikunm ini, dapat disimpulkan bahwa kita bisa membuat program untuk mengontrol LED dengan sensor infrared receiver menggunakan arduinlo. Dengan memahami contoh dari latihan tersebut, kita bisa memahami alur dari program kontrol sensor infrared sehingga kita bisa membuat rangkaian kontrol lainya seperti kontrol kipas atau berbagai infrared project arduino yang lebih kompleks.