

BAB III

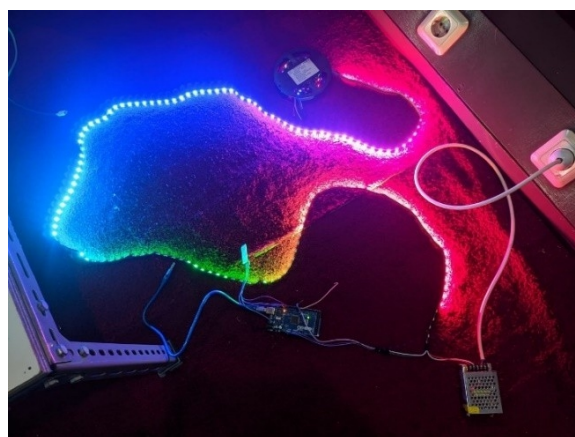
PEMBAHASAN

Pembuatan alat dilaksanakan dengan tahapan studi literatur, merancang perangkat keras yang meliputi rangkaian elektronik, implementasi rangkaian elektronik, merancang perangkat lunak, dan melakukan pengujian. Kemudian menyusun laporan tugas akhir dan. Alur dari pembuatan diperlihatkan pada gambar 2.7

A. Perancangan Perangkat Keras

Konstruksi alat yang dibuat ditunjukkan pada gambar di bawah ini adalah kabel jumper male to male dan kabel jumper male to female dengan komponen utama Arduino mega 2560.

Pada gambar di bawah ini menunjukkan cara kerja kabel jumper yang berada di pin 2 ketika di konsletkan maka akan menunjukkan perubahan dengan berbagai macam warna.



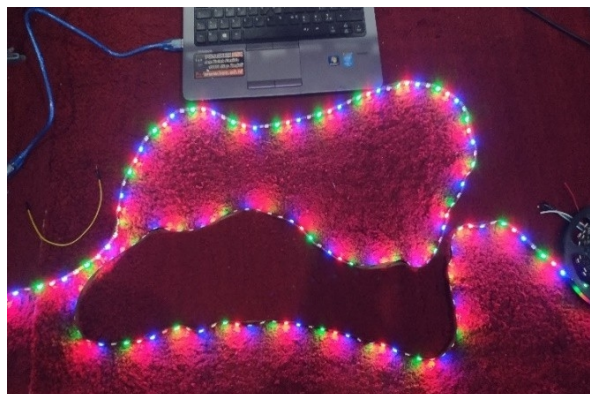
Gambar 3.1 Hasil percobaan tambahan kabel pin 2 RGB

Pada gambar di bawah ini menampilkan cara kerja lampu LED menyala dengan perlahan dan secara bergantian.



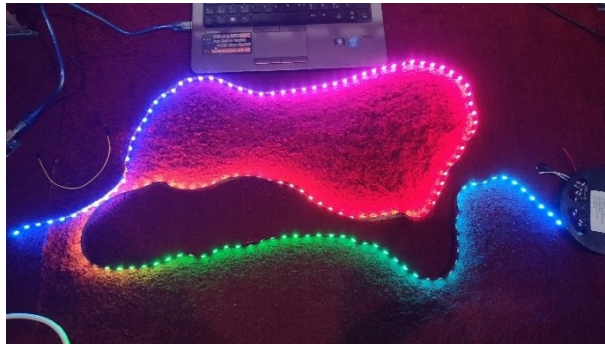
Gambar 3.2 Hasil percobaan lampu menyala dengan lambat

Menampilkan cara kerja lampu LED yang bergerak dengan cepat dan menimbulkan berbagai macam variasi warna.



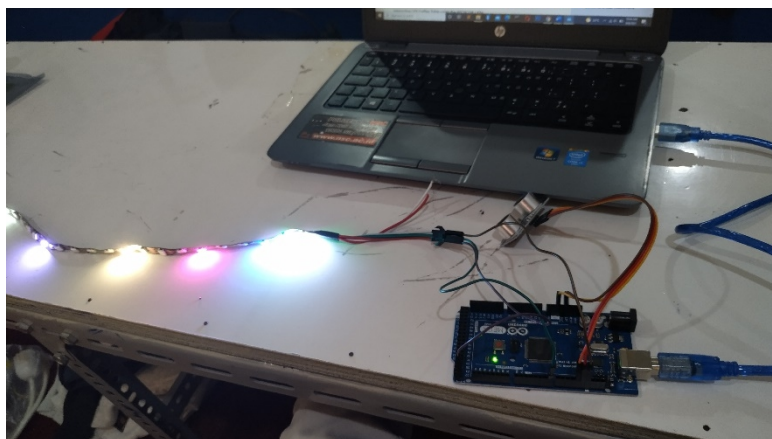
Gambar 3.3 Hasil percobaan lampu menyala RGB

Menampilkan cara kerja lampu LED yang berganti warnan dengan teratur mengikuti aliran tersebut.



Gambar 3.4 Hasil percobaan lampu menyala secara bergantian

Menampilkan cara kerja lampu LED dengan menambahkan sensor ultrasonik



Gambar 3.5 Hasil percobaan dengan penambahan sensor ultrasonik

B. Perancangan Perangkat Lunak

Perancangan perangkat lunak pada gambar di bawah ini adalah codingan lampu menyala dengan lambat dan codingan lampu LED RGB.

```

buttoncycler | Arduino 1.8.15
File Edit Sketch Tools Help

buttoncycler$

#ifdef __AVR__
#include <avr/power.h> // Required for 16 MHz Adafruit Trinket
#endif

// Digital IO pin connected to the button. This will be driven with a
// pull-up resistor so the switch pulls the pin to ground momentarily.
// On a high -> low transition the button press logic will execute.
#define BUTTON_PIN 2

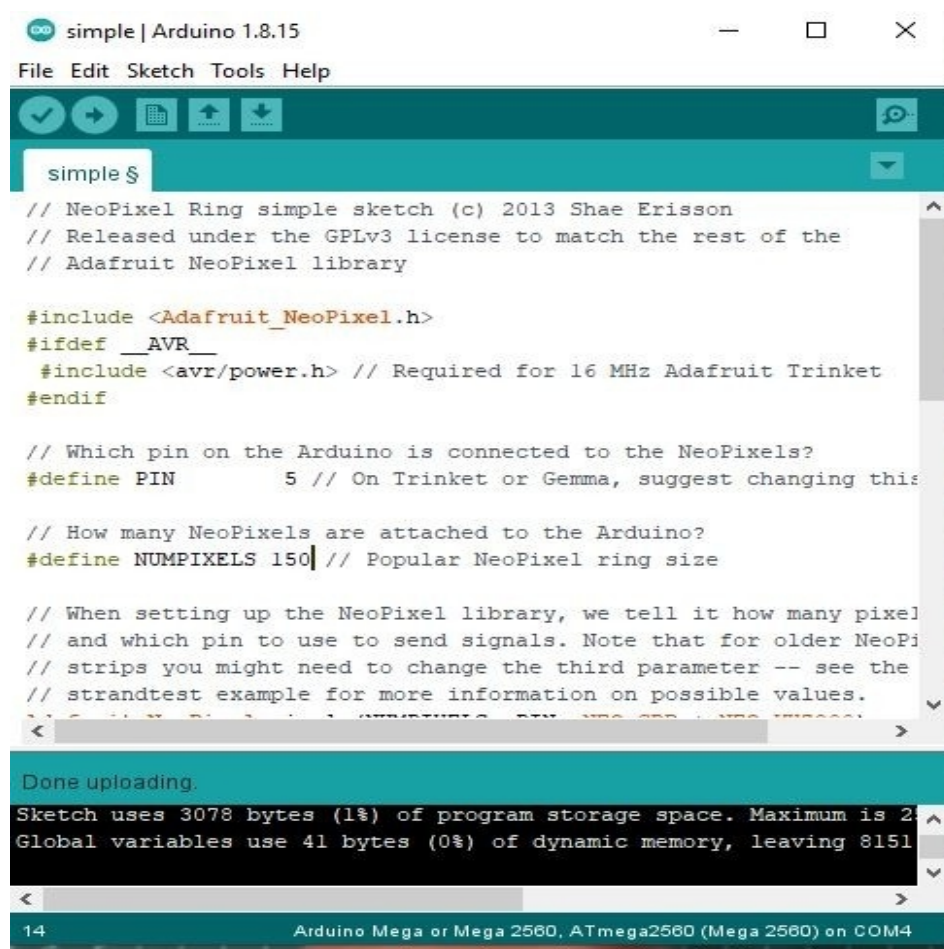
#define PIXEL_PIN 5 // Digital IO pin connected to the NeoPixel strip
#define PIXEL_COUNT 150 // Number of NeoPixels

// Declare our NeoPixel strip object:
Adafruit_NeoPixel strip(PIXEL_COUNT, PIXEL_PIN, NEO_GRB + NEO_KHZ800)
// Argument 1 = Number of pixels in NeoPixel strip
// Argument 2 = Arduino pin number (most are valid)
// Argument 3 = Pixel type flags, add together as needed:
//   NEO_KHZ800 800 KHz bitstream (most NeoPixel products w/WS2812)
//   NEO_GRB     RGB bitstream
//   NEO_RGB     RGB bitstream

Done uploading.
Sketch uses 4842 bytes (1%) of program storage space. Maximum is 262144 bytes.
Global variables use 45 bytes (0%) of dynamic memory, leaving 8147 bytes free.
20 Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM4

```

Gambar 3.6 Hasil coding lampu tambahan kabel pin 2



```

simple | Arduino 1.8.15
File Edit Sketch Tools Help

simple $
// NeoPixel Ring simple sketch (c) 2013 Shae Erisson
// Released under the GPLv3 license to match the rest of the
// Adafruit NeoPixel library

#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
#include <avr/power.h> // Required for 16 MHz Adafruit Trinket
#endif

// Which pin on the Arduino is connected to the NeoPixels?
#define PIN          5 // On Trinket or Gemma, suggest changing this

// How many NeoPixels are attached to the Arduino?
#define NUMPIXELS 150 // Popular NeoPixel ring size

// When setting up the NeoPixel library, we tell it how many pixel
// and which pin to use to send signals. Note that for older NeoPi
// strips you might need to change the third parameter -- see the
// strandtest example for more information on possible values.
// ...

Done uploading.
Sketch uses 3078 bytes (1%) of program storage space. Maximum is 2
Global variables use 41 bytes (0%) of dynamic memory, leaving 8151
14 Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM4

```

Gambar 3.7 Hasil coding lampu menyala dengan lambat


```

RGBWstrandtest | Arduino 1.8.15
File Edit Sketch Tools Help

RGBWstrandtest$
// NeoPixel test program showing use of the WHITE channel for RGBW
// pixels only (won't look correct on regular RGB NeoPixel strips)

#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
#include <avr/power.h> // Required for 16 MHz Adafruit Trinket
#endif

// Which pin on the Arduino is connected to the NeoPixels?
// On a Trinket or Gemma we suggest changing this to 1:
#define LED_PIN 5

// How many NeoPixels are attached to the Arduino?
#define LED_COUNT 15

// NeoPixel brightness, 0 (min) to 255 (max)
#define BRIGHTNESS 50 // Set BRIGHTNESS to about 1/5 (max = 255)

// Declare our NeoPixel strip object:
Adafruit_NeoPixel strip = Adafruit_NeoPixel(LED_COUNT, LED_PIN, NEO_GRB + NEO_WHITE);

Done uploading.
Sketch uses 5324 bytes (2%) of program storage space. Maximum is 2
Global variables use 41 bytes (0%) of dynamic memory, leaving 8151

```

Gambar 3.8 Hasil coding lampu RGB

```

strandtest_wheel | Arduino 1.8.15
File Edit Sketch Tools Help

strandtest_wheel$
#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
#include <avr/power.h>
#endif

#define PIN 5

// Parameter 1 = number of pixels in strip
// Parameter 2 = Arduino pin number (most are valid)
// Parameter 3 = pixel type flags, add together as needed:
//   NEO_KHZ800  800 KHz bitstream (most NeoPixel products w/WS2812)
//   NEO_KHZ400  400 KHz (classic 'v1' (not v2) FLORA pixels, WS2812)
//   NEO_GRB     Pixels are wired for GRB bitstream (most NeoPixel)
//   NEO_RGB     Pixels are wired for RGB bitstream (v1 FLORA pixels)
//   NEO_RGBW    Pixels are wired for RGBW bitstream (NeoPixel RGBW)
Adafruit_NeoPixel strip = Adafruit_NeoPixel(15, PIN, NEO_GRB + NEO_RGBW);

// IMPORTANT: To reduce NeoPixel burnout risk, add 1000 uF capacit
// pixel power leads, add 300 - 500 Ohm resistor on first pixel's

Done uploading.
Sketch uses 4056 bytes (1%) of program storage space. Maximum is 2
Global variables use 41 bytes (0%) of dynamic memory, leaving 8151

```

Gambar 3.9 Hasil coding lampu menyala secara bergantian