

PERCOBAAN I

BASIC INPUT/OUTPUT

1.1 TUJUAN PERCOBAAN

Tujuan dari praktikum bab 1 ini adalah praktikan mampu memahami konsep dasar penggunaan input/output pada Arduino Uno, mampu membuat program untuk mengeluarkan output dan mengakses input dari tiap-tiap port, dan mampu mendesain dan mengimplementasikan sebuah sistem sederhana yang melibatkan input dan output pada simulator SimulIDE.

1.2 PERANGKAT YANG DIGUNAKAN

Perangkat untuk simulasi:

- PC/Laptop
- Software SimulIDE (Untuk Simulasi)
- Arduino IDE

Perangkat untuk non simulasi:

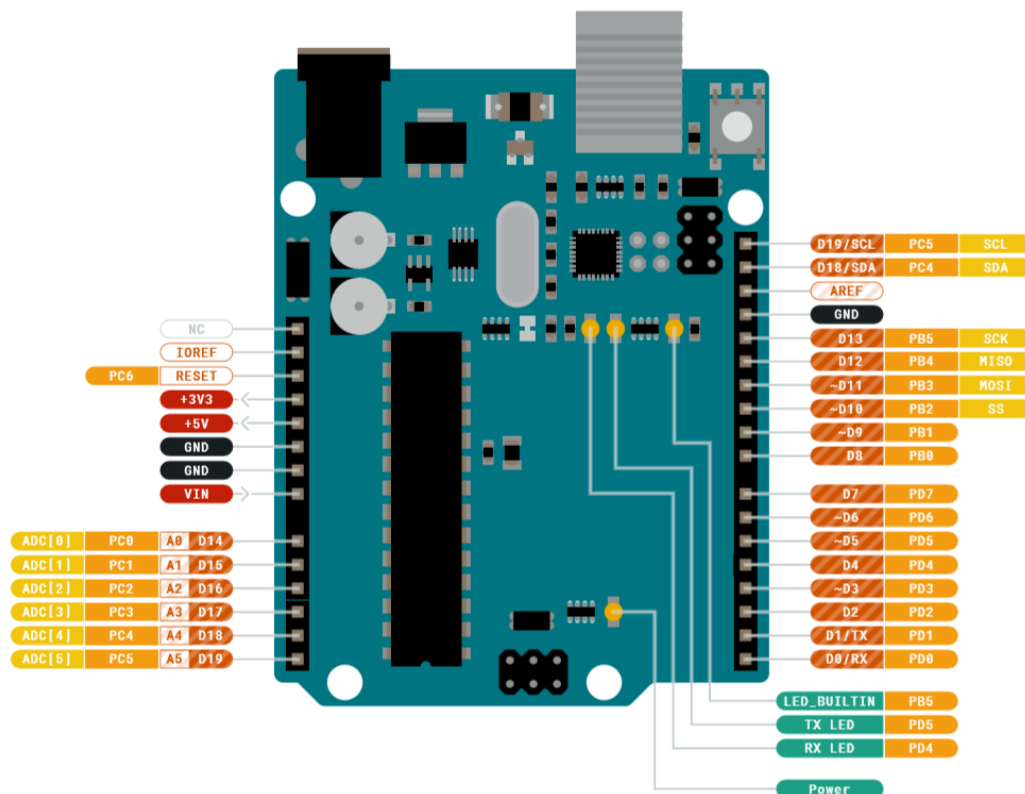
- Perangkat lengkap Arduino UNO
- LED dan resistor
- Push button atau saklar

1.3 DASAR TEORI

Sebelum melakukan percobaan pada praktikum ini, ada beberapa teori tentang hardware dan software yang digunakan yang harus dipahami terlebih dahulu oleh praktikan. Beberapa teori tersebut berkaitan dengan Arduino Uno dan simulator SimulIDE.

ARDUINO UNO

Arduino Uno merupakan sebuah modul rangkaian elektronik yang bersifat open source. Di dalam *Arduino Uno* terdapat chip mikrokontroler yang merupakan komponen utama yang dibuat oleh perusahaan ATMEL. Chip mikrokontroler yang digunakan adalah Atmega328 dari jenis AVR. Bentuk modul dan *pinout* Arduino Uno ditunjukkan dalam Gambar 1.1. Sedangkan spesifikasi teknis mengenai arduino uno akan diuraikan pada Tabel 1.1.



Gambar 1.1 *Arduino Uno*

Tabel 1.1 Spesifikasi *Arduino Uno*

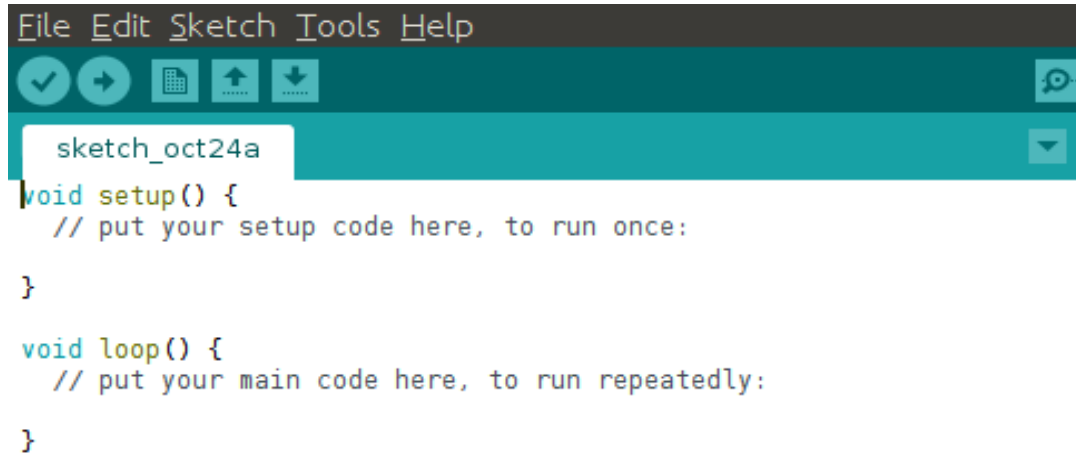
Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (disarankan)	7-12V
Input Voltage (batas)	6-20V
Digital I / O	Pin 14 (6 memberikan output PWM)
Analog Input	Pin 6
DC Current per I / O	Pin 40 mA
DC Current for 3.3v	Pin 50 Ma
Flash Memory 32 KB (Atmega328)	0,5 KB digunakan oleh bootloader
SRAM	2 KB (Atmega328)
EEPROM	1 KB (Atmega328)
Clock Speed	16 MHz

Untuk dapat menggunakan *Arduino Uno* dalam sebuah sistem, ada dua bagian yang harus dipersiapkan:

- 1) *Hardware* modul *Arduino Uno* yang berisikan pin input/output (I/O).

2) Software Arduino IDE.

Software Arduino IDE digunakan untuk menulis program, mengkompilasi program menjadi kode biner dan mengunggah hasil kompilasi program tersebut ke dalam memori *flash* mikrokontroler yang ada di modul *Arduino Uno*. Bahasa yang digunakan dalam *Arduino IDE* adalah bahasa C++ dengan beberapa penyesuaian. Tampilan program *Arduino IDE* ini ditunjukkan dalam Gambar 1.2.



Gambar 1.2 Tampilan Arduino IDE

Struktur pemrograman pada *Arduino IDE* secara garis besar terdiri atas dua bagian utama, yaitu:

- 1) ***void setup()***: Bagian ini hanya akan dieksekusi satu kali pada awal program dijalankan (modul *Arduino Uno* dinyalakan atau direset). Bagian ini berisikan perintah inisialisasi untuk fitur-fitur yang akan digunakan.
- 2) ***void loop()***: Bagian ini akan dieksekusi secara berulang-ulang dan terus-menerus sampai *Arduino Uno* dimatikan. Bagian ini berisi program utama yang menjalankan fitur-fitur yang diinginkan dan juga aplikasi yang kita inginkan.

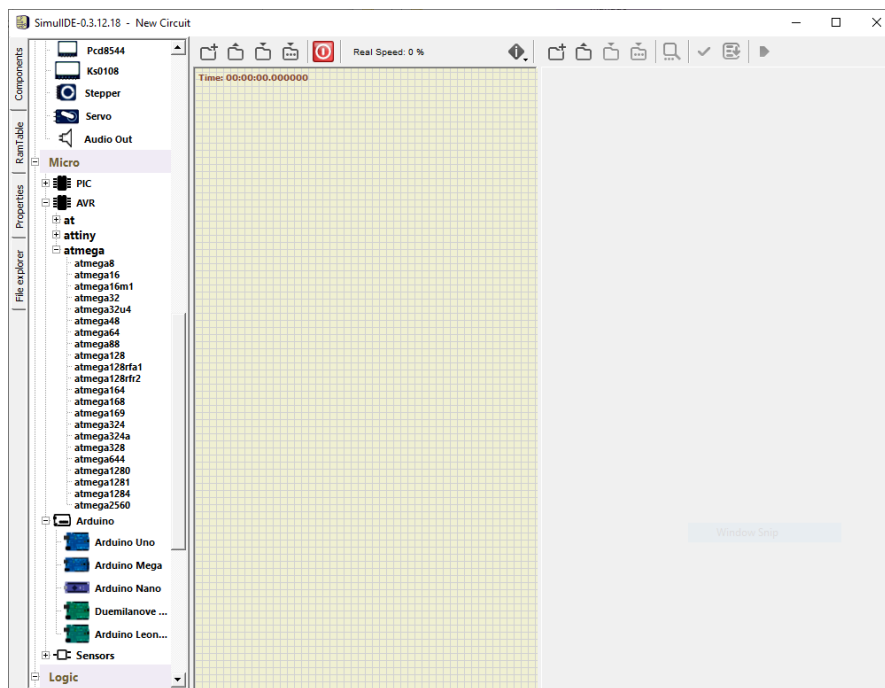
Untuk menggunakan input output pada *Arduino Uno* diperlukan beberapa *syntax* dasar. *Syntax* ini meliputi *syntax* untuk inisialisasi yang berfungsi menentukan sebuah pin sebagai input atau output dan *syntax* eksekusi yang berfungsi untuk membaca data dari pin atau mengeluarkan data pada pin, baik digital maupun analog. Beberapa *syntax* ini diuraikan dan dijelaskan pada Tabel 1.2.

Tabel 1.2 *Syntax* Input & Output beserta fungsi masing-masing *syntax*

<i>Syntax</i>	Kegunaan	Parameter	Nilai kembalian
---------------	----------	-----------	-----------------

<i>pinMode(pin, mode)</i>	Menentukan sebuah pin sebagai input/output	1) Nomor pin 2) Mode (I/O)	-
<i>digitalRead (pin)</i>	Membaca data digital pada pin	Nomor pin	High/low
<i>analogRead (pin)</i>	Membaca data analog pada pin	Nomor pin	0-1023
<i>digitalWrite (pin,value)</i>	Memberikan nilai digital pada pin	1) Nomor pin 2) Nilai (high/low)	-
<i>AnalogWrite(pin,value)</i>	Memberikan nilai analog pada pin	1) Nomor pin 2) Nilai (high/low)(0 – 1023)	-

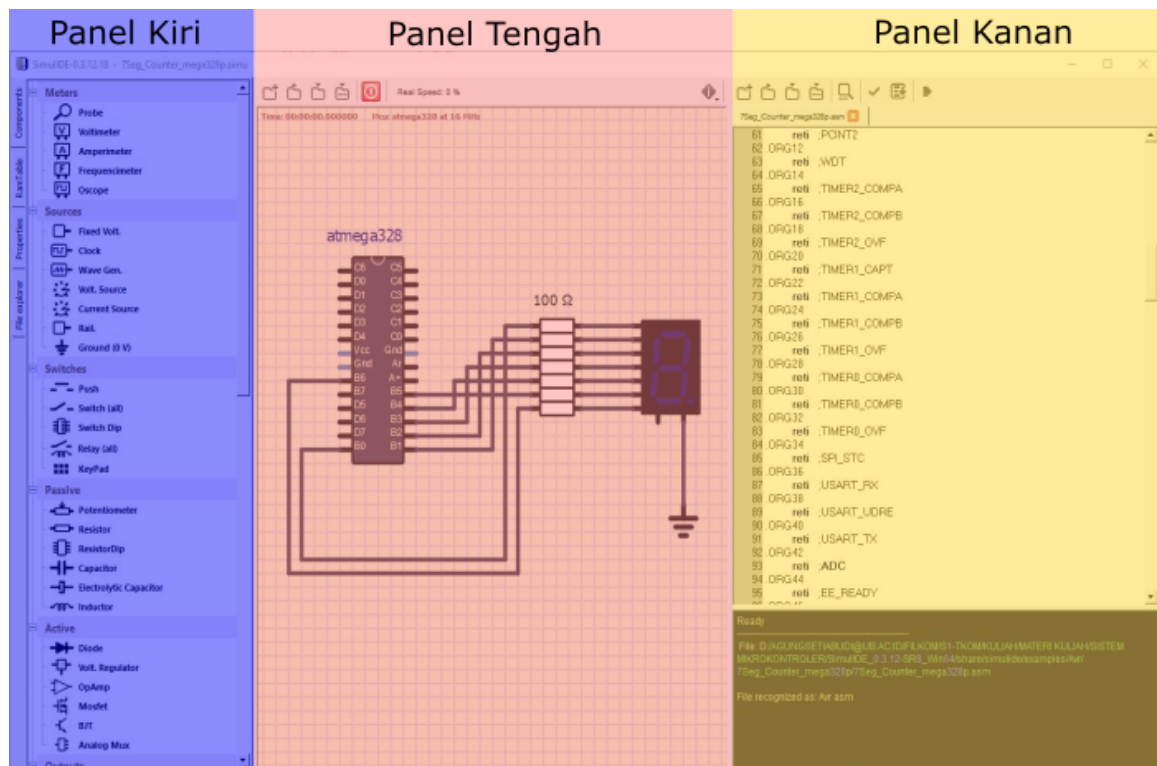
Dalam praktikum ini, *Arduino Uno* akan dihubungkan dengan input dan output pada simulator SimulIDE. SimulIDE adalah sebuah simulator rangkaian elektronik yang bersifat *real time*. Simulator ini juga mendukung penggunaan beberapa mikrokontroler, seperti AVR, Arduino, dan PIC. Tampilan *window* SimulIDE ini ditunjukkan dalam Gambar 1.3.



Gambar 1.3 Tampilan aplikasi SimulIDE

Aplikasi SimulIDE memiliki tiga buah panel utama seperti ditunjukkan dalam Gambar 1.4. Yang pertama adalah panel kiri yang berisi empat *tab*, yaitu *Components*, *Ram table*, *properties*, dan *File explorer*. Tab *Components* berisi komponen-komponen yang bisa digunakan dalam simulasi. Komponen-komponen ini terdiri atas:

- 1) **Meters:** Berisi komponen-komponen yang digunakan untuk mengukur besaran listrik seperti voltmeter dan amperemeter.
- 2) **Sources:** Berisi sumber tegangan atau sumber arus yang bisa digunakan untuk memberi catu daya pada rangkaian elektronik yang kita simulasikan. Di bagian ini juga terdapat sumber *clock* dan *wave generator*.
- 3) **Switches:** Berisi berbagai macam saklar mulai dari *push button*, *relay*, sampai *keypad*.
- 4) **Passive:** Berisi komponen-komponen pasif seperti resistor, kapasitor, dan induktor.
- 5) **Active:** Berisi komponen-komponen aktif seperti dioda, transistor, op-amp, dan lain-lain.
- 6) **Outputs:** Berisi komponen-komponen yang bisa digunakan sebagai output, seperti LED, LED matriks, 7 segment, LCD, motor stepper, audio, dan lain-lain.
- 7) **Micro:** Berisi berbagai macam mikrokontroler (AVR, Arduino, dan PIC)
- 8) **Logic:** Berisi komponen-komponen digital, seperti gerbang logika dan juga IC yang terdiri atas beberapa gerbang logika di dalamnya.
- 9) **Other:** Berisi hal-hal lain yang bisa memudahkan simulasi, seperti text dan package.



Gambar 1.4 Panel pada SimulIDE

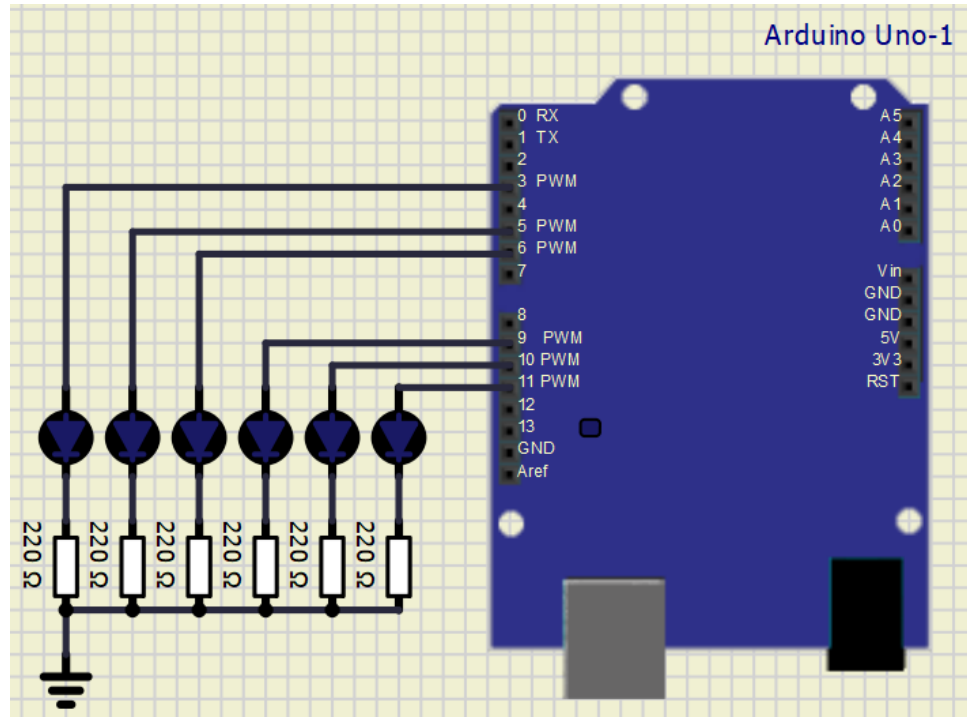
Panel yang kedua adalah panel tengah. Panel ini digunakan untuk meletakkan komponen-komponen yang ingin kita simulasikan. Semua komponen yang kita simulasikan bisa dirangkai dan dihubungkan dengan komponen-komponen lain di panel ini. Di panel ini juga kita bisa berinteraksi dengan input dan output dari rangkaian yang kita buat. Panel yang terakhir adalah panel kanan. Panel ini bisa digunakan sebagai *code editor*. Adapun source code yang bisa dituliskan di sini adalah GcBasic, Arduino, PIC asm and AVR asm. Pada praktikum ini, penulisan kode program akan dilakukan secara langsung dalam simulator ini.

1.4 PROSEDUR PERCOBAAN

PERCOBAAN 1: PENGGUNAAN PIN SEBAGAI OUTPUT

Persiapan

- Buatlah rangkaian pada SimulIDE seperti pada Gambar 1.5 berikut ini.



Gambar 1.5 Rangkaian percobaan 1.5 pada SimulIDE

- Tuliskan program 1.1 di bawah ini dan *compile*.

Program 1.1

```
1 char pinsCount = 6;
2 int pins[] = {3,5,6,9,10,11};
3
4 void setup() {
5   for (int i = 0; i < pinsCount; i++){
6     pinMode(pins[i], OUTPUT);
7   }
8 }
9
10 void loop() {
11   for (int i = 0; i < pinsCount; i++){
12     digitalWrite(pins[i], HIGH);
13     delay(1000);
14     digitalWrite(pins[i], LOW);
15   }
16 }
```

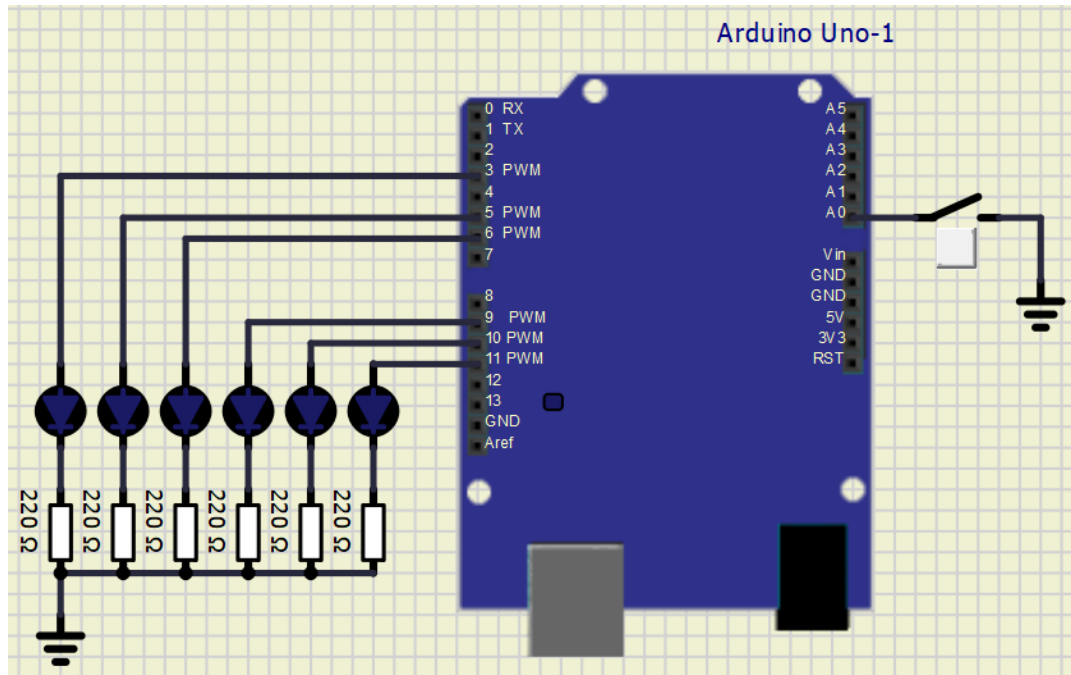
- Setelah program berhasil dikompilasi, tekan tombol *upload*.
- Jalankan simulasi dengan cara menekan tombol Power pada SimulIDE.

ANALISIS 1

1. Ceritakan bagaimana nyala LED setelah simulasi dijalankan!
2. Jelaskan kegunaan tiap-tiap baris source code!

PERCOBAAN 2: PENGGUNAAN PIN SEBAGAI INPUT DAN OUTPUT

- Buatlah rangkaian pada SimulIDE seperti pada Gambar 1.6 berikut ini.



Gambar 1.6 Rangkaian percobaan 1.2 pada SimulIDE

- Tuliskan program 1.2 di bawah ini dan *compile*.

Program 1.2

```
1#define tombol A0
2char pinsCount = 6;
3int pins[] = {3,5,6,9,10,11};
4int val = 0;
5
6void setup() {
7    for (int i = 0; i < pinsCount; i++){
8        pinMode(pins[i], OUTPUT);
9    }
10    pinMode(tombol, INPUT_PULLUP);
11}
12
13void loop() {
14    val = digitalRead(tombol);
15    if (val == HIGH){
16        for (int i = 0; i < pinsCount; i++){
17            for (int j = 0; j <= 255; j++){
18                analogWrite(pins[i], j);
19                delay(5);
20            }
21            for (int j = 255; j >= 0; j--){
22                analogWrite(pins[i], j);
```

```

23         delay(5);
24     }
25 }
26 }
27 else if (val == LOW){
28     for (int i = pinsCount - 1; i >= 0; i--){
29         for (int j = 0; j <= 255; j++){
30             analogWrite(pins[i], j);
31             delay(5);
32         }
33         for (int j = 255; j >= 0; j--){
34             analogWrite(pins[i], j);
35             delay(5);
36         }
37     }
38 }
39 }

```

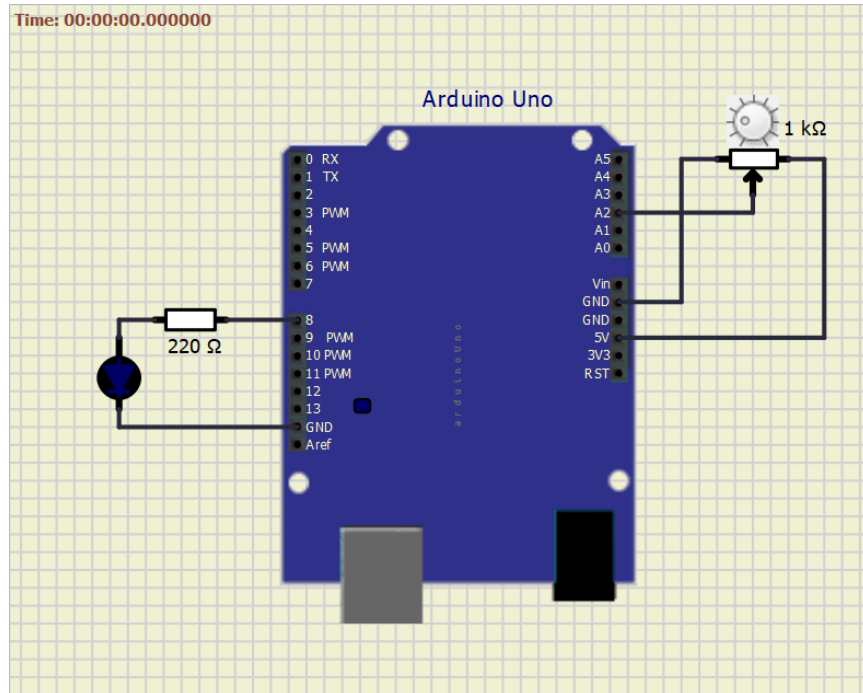
- Setelah program berhasil dikompilasi, tekan tombol *upload*.
- Jalankan simulasi dengan cara menekan tombol Power pada SimulIDE.

ANALISIS 2

1. Ceritakan bagaimana nyala LED setelah simulasi dijalankan!
2. Apa yang membedakan percobaan 1.2 dari percobaan 1.1 sebelumnya? Jelaskan!
3. Tekan tombol yang terhubung ke PIN A0 dan ceritakan bagaimana nyala LED pada saat *switch* ditekan.
4. Jelaskan kegunaan tiap-tiap baris source code!

PERCOBAAN 3: ANALOG DIGITAL CONVERTER

- Buatlah rangkaian pada SimulIDE seperti pada Gambar 1.7 berikut ini.



Gambar 1.7 Rangkaian percobaan 1.3 pada SimulIDE

Program 1.3

```

1 void setup() {
2   Serial.begin(9600);
3 }
4
5 void loop() {
6   int sensorValue = analogRead(A2);
7   Serial.println(sensorValue);
8   digitalWrite(8, HIGH);
9   delay(sensorValue);
   digitalWrite(8, LOW);
   delay(sensorValue);
}

```

- Setelah program berhasil dikompilasi, tekan tombol *upload*.
- Putar potentiometer ke kanan dan kekiri
- Lihat Serial Monitor pada Arduino IDE dan durasi led *blinking*.

Pin A2 merupakan pin analog dan memiliki fitur *Analog to digital* converter yang melakukan konversi signal analog yang masuk antara (0-5V) menjadi *range* numbers 0-1023.

1.5 TUGAS

1. Pada percobaan 1.2 ada bagian kode yang ditulis berulang. Buatlah fungsi pada program percobaan 1.2 untuk meminimalisasi penulisan kode yang berulang!
2. Pada percobaan 1.2, ketika tombol ditekan program masih harus menyelesaikan eksekusi program sebelumnya sehingga ada jeda waktu di situ. Rancang dan buatlah sebuah program dan rangkaianannya seperti skenario percobaan 1.2, yang jika tombol ditekan maka *Arduino uno* akan segera mengubah arah nyala LED.
3. Pada percobaan 1.3, jelaskan apa yang dimaksud dengan ADC dan bagaimana prinsip kerja ADC pada arduino
4. Pada percobaan 1.3, jelaskan mengapa angka pada potentiometer menunjukkan nilai 0-1023?

1.6 KESIMPULAN

Berdasarkan percobaan yang telah dilakukan, maka tuliskan kesimpulan percobaan di tempat yang telah disediakan di bawah ini: