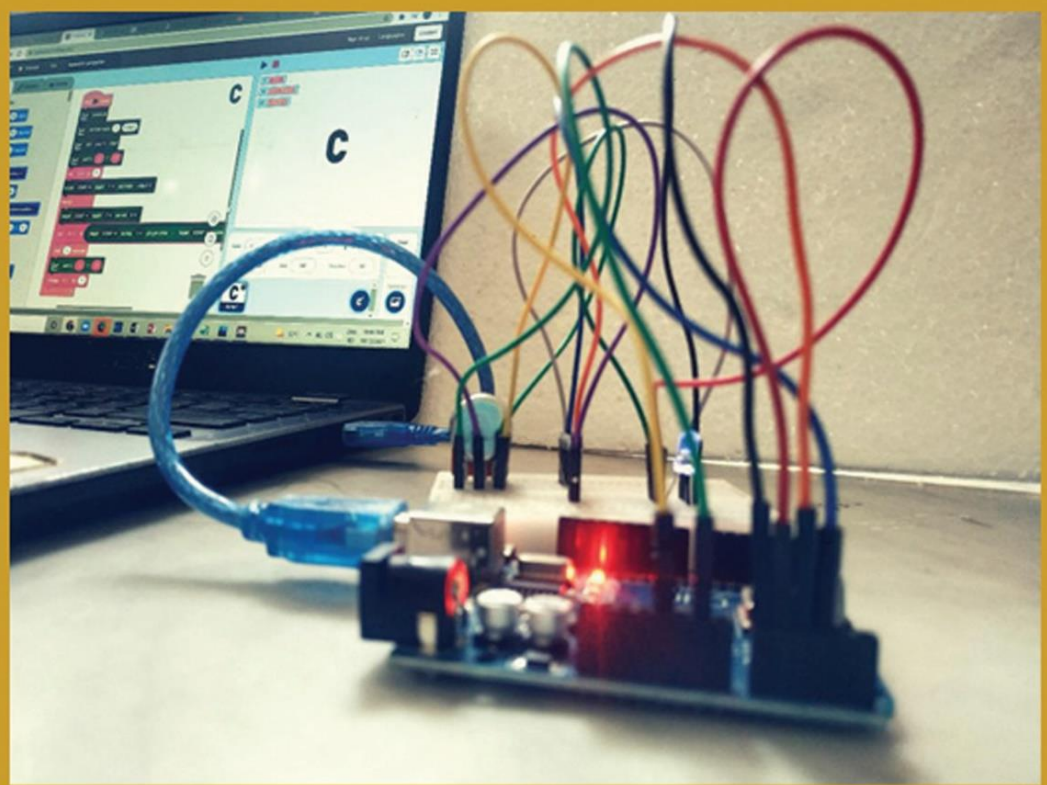




Penggunaan Arduino Uno dan Common-Coding

Pada Percobaan Percobaan Fisika Materi Kelistrikan



Elisabeth Pratidhina
Dadan Rosana
Heru Kuswanto

Penggunaan Arduino Uno dan Common-Coding pada Percobaan Fisika Materi Kelistrikan

Oleh

Elisabeth Pratidhina

Dadan Rosana

Heru Kuswanto



Cipta Media Nusantara
2021

Penggunaan Arduino Uno dan Common-Coding pada Percobaan Fisika Materi Kelistrikan

Penulis : Elisabeth Pratidhina
: Dadan Rosana
: Heru Kuswanto
Editor : Musliman
Layout : Khoyrul Umam
Cover : Mohammad Nasir

Diterbitkan dan Dicitak Oleh:

Cipta Media Nusantara (CMN), 2021

Anggota IKAPI: 270/JTI/2021

Alamat : Jl. Jemurwonosari 1/39, Wonocolo, Surabaya

Email : ciptapublishing@gmail.com

Web : www.ciptapublishing.com

ISBN : 978-623-5647-08-1

Cetakan Pertama: Oktober 2021

Copyright © 2021 Cipta Media Nusantara

Ketentuan Pidana Pasal 112-119

Undang-undang Nomor 28 Tahun 2014 Tentang Hak Cipta.

Dilarang keras menerjemahkan, memfotokopi, atau memperbanyak sebagian atau seluruh isi buku ini tanpa izin tertulis dari penerbit

KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Kuasa atas berkat dan karunia yang diberikan sehingga tim penulis dapat menyelesaikan buku berjudul “Penggunaan Arduino Uno dan Common-Coding dalam Praktikum Fisika Materi Kelistrikan”. Penulis mengucapkan terimakasih kepada beberapa pihak diantaranya:

1. Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi yang telah memberikan dukungan pendanaan penelitian dan penulisan buku melalui Hibah Penelitian Disertasi Doktor dengan nomor kontrak:
134 /SP2H/LT/DRPM/2021 &
T/4.2.21/UN34.21/PT.01.03/2021.
2. Universitas Negeri Yogyakarta dan Universitas Katolik Widya Mandala Surabaya atas dukungan fasilitas yang diberikan kepada penulis.

Praktikum memiliki peranan esensial dalam pembelajaran fisika. Arduino Uno merupakan sebuah mikrokontroler dengan harga terjangkau yang berpotensi untuk digunakan dalam praktikum fisika. Pada umumnya, Arduino diprogram menggunakan pemrograman berbasis C++. Buku ini menjelaskan alternatif pemrograman Arduino Uno dengan menggunakan pemrograman berbasis blok bernama Common-Coding. Dengan Common-Coding, pengguna hanya perlu melakukan *drag and drop* pada blok untuk mengkonstruksi sebuah program tanpa perlu menghafal sintaks pengkodean.

Pembahasan pada buku ini diawali dengan pengenalan Arduino Uno serta bagaimana menggunakan Common-Coding untuk memprogram Arduino Uno. Setelah itu, dibahas tentang

aplikasi Arduino dalam praktikum fisika, khususnya tentang kelistrikan. Contoh topik-topik praktikum fisika menggunakan Arduino dan Common-Coding juga dijelaskan untuk memudahkan pendidik ketika akan mengimplementasikannya dalam pembelajaran fisika.

Diharapkan buku ini dapat memberikan kontribusi bagi inovasi pembelajaran fisika, sehingga Praktikum fisika dengan Arduino dan Common-Coding dapat diimplementasikan pada pembelajaran jarak jauh maupun tatap muka. Penulis menyadari bahwa buku ini masih belum sempurna. Oleh sebab itu, penulis sangat terbuka menerima masukan dari para pembaca.

Yogyakarta, 10 Oktober 2021

Penulis

DAFTAR ISI

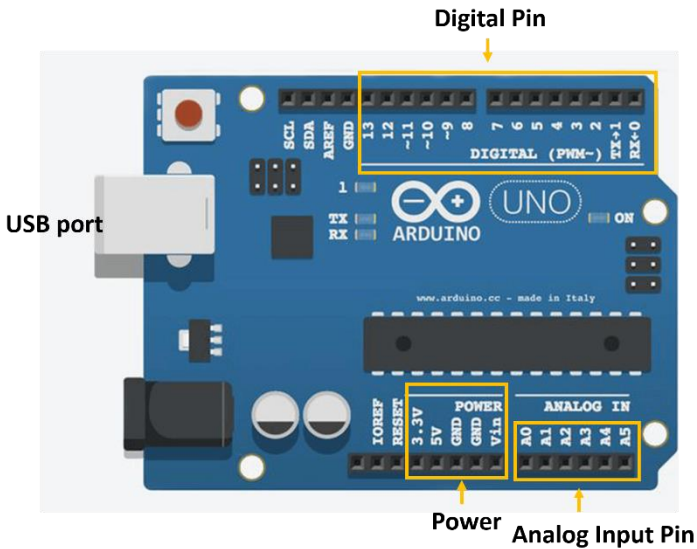
Kata Pengantar	iii
Daftar Isi	v
BAB 1 PENGENALAN ARDUINO UNO	1
A. Bagian-Bagian Arduino Uno	2
B. Menghubungkan Arduino Uno Ke Komputer	3
C. <i>Breadboard</i> dan Diagram Rangkaian Listrik	6
BAB 2 COMMON-CODING	15
A. Pengenalan Common-Coding	15
B. Menghubungkan Arduino ke Common Coding	16
C. Langkah menyusun script pada Common-Coding	21
D. Langkah mendeklarasi variabel	24
E. Menyimpan Proyek dan Membukanya Kembali	27
BAB 3 APLIKASI ARDUINO DALAM PERCOBAAN FISIKA	29
A. Pengukuran Tegangan	29
B. Pengukuran Arus	33
C. Pengukuran Hambatan	36
D. Menggunakan Arduino Sebagai Sumber Tegangan DC	38
BAB 4 CONTOH PERCOBAAN FISIKA MENGGUNAKAN ARDUINO	41
A. Percobaan Hukum Ohm (Hubungan V-I pada Konduktor)	41

B. Percobaan Hukum Ohm (Mengidentifikasi Faktor Penentu Hambatan pada Konduktor)	47
C. Percobaan Mengidentifikasi <i>Ohmic</i> dan <i>Non-Ohmic</i> Material	54
D. Percobaan Menentukan Hambatan Pengganti Seri dan Paralel	61
E. Percobaan Hukum Kirchhoff	67
F. Percobaan Pengisian dan Pengosongan Kapasitor (Rangkaian <i>RC</i>)	74
Daftar Pustaka	79
Glosarium	81
Indeks	83
Catatan	84

BAB 1

PENGENALAN ARDUINO UNO

Arduino Uno merupakan salah satu jenis papan mikro-kontroler yang dikembangkan oleh Arduino.cc. Arduino dapat dikoneksikan ke komputer dengan kabel USB dan diprogram dengan menggunakan Software Arduino (IDE) yang mendukung bahasa pemrograman C dan C++ ataupun dengan software lain seperti *Scratch for Arduino* atau *Common-Coding* yang menggunakan bahasa pemrograman berbasis *block* /gambar (Kadir, 2016; Pratidhina, Rosana, Kuswanto, & Dwardaru, 2021). Arduino Uno dapat digabungkan dengan berbagai sensor pengukuran besaran fisika sehingga sangat cocok untuk mendukung kegiatan eksperimen fisika (Atkin, 2018; Galeriu, 2018; Kubínová & Šlégr, 2015; Yasin, Prima, & Sholihin, 2018). Sebelum kita memanfaatkan Arduino Uno dalam pembelajaran fisika, kita perlu kenali beberapa bagian-bagian dari Arduino Uno.



Gambar 1.1. Papan Arduino Uno

A. Bagian-Bagian Arduino Uno

1. Ground pins (GND)

GND pada papan Arduino merupakan pin untuk menghubungkan ke ground. GND sangat penting pada papan Arduino untuk mengukur dan mengatur tegangan. Pada dasarnya, tegangan adalah beda potensial antara dua titik, kita dapat menggunakan titik GND dan titik yang lainnya. Rangkaian harus dihubungkan pada GND, sehingga semua tegangan yang terbaca bisa saling dibandingkan dan nilainya relevan karena memiliki titik referensi yang sama. Jika tidak ada *common ground*, tidak ada artinya kita membandingkan hasil pembacaan 3,3 V dan 5 V.

2. Sumber Tenaga Arduino

Supaya papan Arduino dapat menjalankan kerjanya, papan ini memerlukan energi listrik. Sumber energi listrik papan Arduino dapat diperoleh dengan menghubungkan Arduino ke komputer dengan menggunakan kabel USB. Selain itu, jack DC power juga dapat digunakan untuk memberikan aliran listrik ke papan Arduino (tegangannya 7-12 V). Jika ingin menggunakan sumber listrik ini, maka kita perlu memanfaatkan pin V_{in} . Ketika menggunakan V_{in} , kita perlu menggunakan GND dengan benar, pastikan perangkat sumber listrik eksternal juga terhubung ke GND.

3. Digital Pin

Arduino Uno memiliki 14 digital pin, yang diberi label angka 1-13. Pin digital merupakan pin *binary* yang hanya memiliki 2 keadaan yaitu LOW (0) atau HIGH (1). Pin digital dapat diatur menjadi output dan input.

Ketika pin digital digunakan untuk mode input, kita akan menggunakannya untuk membaca data. Data yang terbaca juga hanya ada 2 jenis, yaitu LOW (0) dan HIGH (1). Saat dipakai sebagai INPUT, pin digital Arduino Uno akan membaca

tegangan kurang dari 0,8 V sebagai LOW (1) dan tegangan yang lebih dari 2 V akan terbaca sebagai HIGH (1).

Pada saat menjadi OUTPUT, LOW (0) berarti tegangan yang dikeluarkan dari pin adalah 0V. Sedangkan HIGH berarti Vcc atau 5V pada Arduino Uno. Namun demikian, ada beberapa pin digital yang merupakan pin PWM (*pulse width modulation*), dimana tegangan output pin PWM tersebut dapat diatur antara 0-5V. Pada Arduino Uno, pin PWM memiliki tanda ~ (pin PWM pada Arduino Uno adalah no ~11, ~10, ~9, ~6, ~5, dan ~3).

4. Pin Analog

Terdapat 6 pin analog pada papan Arduino Uno, pin tersebut diberi label A0-A5. Pin analog ini hanya dapat digunakan untuk pembacaan nilai tegangan. Pin analog berguna untuk membaca nilai masukan yang bukan hanya HIGH atau LOW, namun bisa nilai di antaranya.

Cara kerja pin analog adalah dengan menangkap nilai tegangan yang masuk. Nilai tegangan yang bisa terbaca adalah 0-5V. Kemudian ADC (*Analog Digital Converter*) akan mengubah nilai analog tersebut menjadi nilai yang dapat dipahami oleh program Arduino yaitu nilai digital.

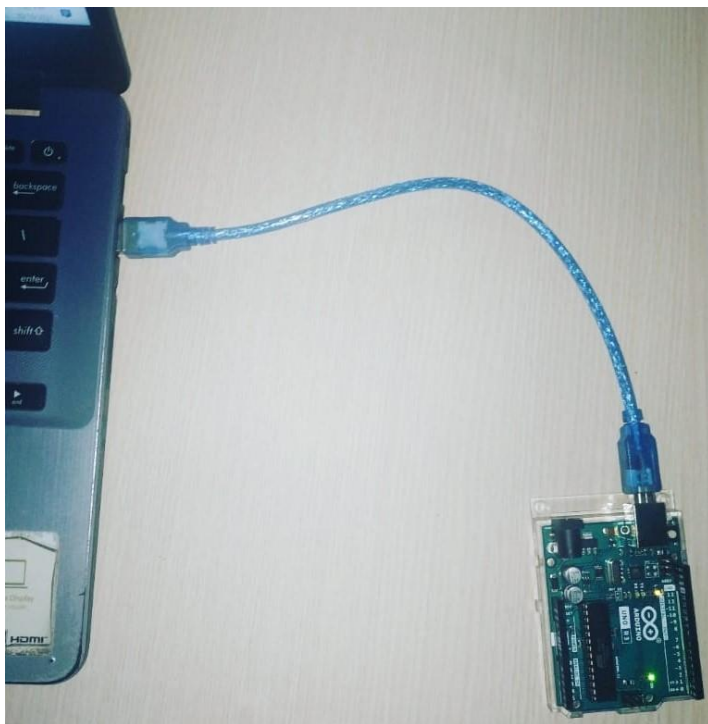
Papan Arduino Uno memiliki 10 bit ADC. Dengan 10 bit tersebut, maka resolusinya adalah $2^{10}=1024$. Sehingga nilai yang diperoleh dari pin analog input adalah di antara 0 dan 1023. Nilai tegangan sesungguhnya (0-5V) dikonversi menjadi bilangan di antara 0-1023 oleh ADC. Misalnya, kita membaca tegangan suatu rangkaian yang bernilai 2,5 V; maka nilai pembacaan pin analog adalah $2,5V / 5V \times 1023 = 512$.

B. Menghubungkan Arduino Uno Ke Komputer

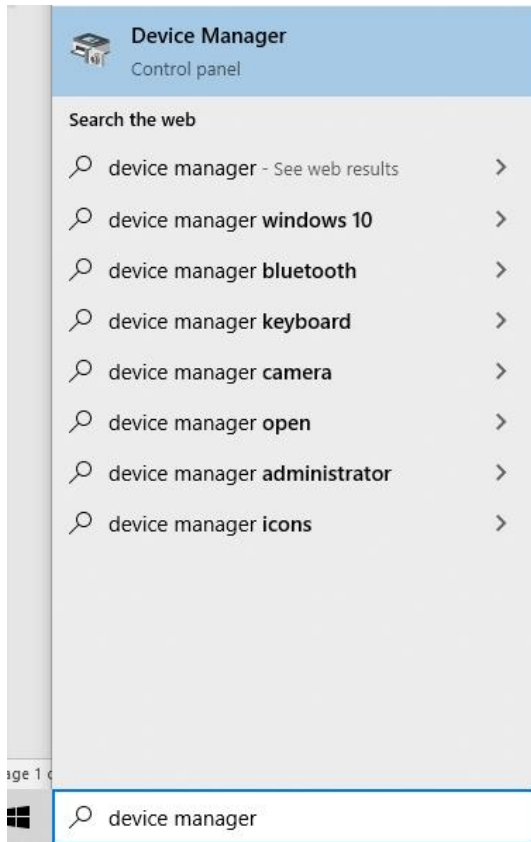
Supaya komputer dapat mengenali Arduino Uno, diperlukan driver Arduino Uno yang terinstall pada komputer. Jika kita menggunakan papan Arduino Uno original, driver sudah

menyatu dengan program Arduino yang dapat didownload di <https://www.arduino.cc/en/software>. Jika Uno yang digunakan adalah versi *clone* (bukan original), kadang diperlukan driver yang berbeda. Pada buku ini, penulis menggunakan Arduino Uno versi original.

Setelah driver terinstall, selanjutnya Arduino Uno harus dihubungkan ke komputer dengan menggunakan kabel USB serial. Untuk mengetahui apakah Arduino sudah terbaca oleh komputer, kita dapat membuka Device Manager. Pada Windows 10, Device manager dapat dibuka dengan mengetikkan 'Device Manager' pada kolom Search.

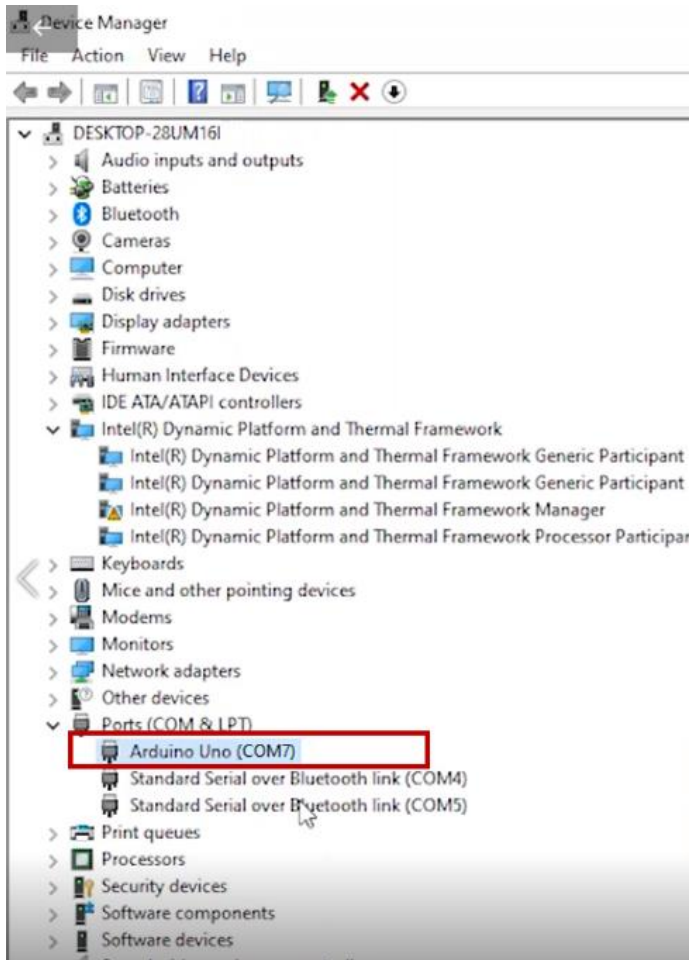


Gambar 1.2. Koneksi Arduino Uno ke komputer dengan menggunakan kabel serial USB



Gambar 1.3. Membuka Device Manager

Pada device manager, terdapat bagian Ports. Bila pada Port sudah terdeteksi 'Arduino Uno', berarti Arduino sudah berhasil dihubungkan dengan komputer. Nomor COM dimana Arduino terhubung perlu diperhatikan karena Informasi ini diperlukan ketika akan mengirimkan perintah ke Arduino melalui pemrograman. Pada contoh di Gambar, Arduino terhubung pada COM7.



Gambar 1.4. Arduino Uno telah dapat terdeteksi oleh komputer dan terkoneksi melalui COM 7.

C. Breadboard dan Diagram Rangkaian Listrik

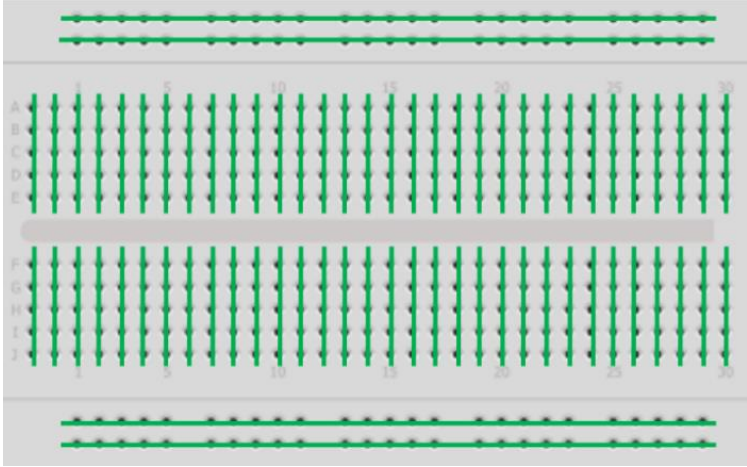
Breadboard merupakan papan yang biasa digunakan untuk merangkai komponen listrik sementara. *Breadboard* biasa terbuat dari plastic dan terdapat banyak lubang pada salah satu sisinya. Lubang tersebut merupakan tempat untuk meletakkan komponen seperti resistor, kapasitor, LED, diode, dll. Lubang-lubang *breadboard* diatur sedemikian rupa

sehingga membentuk pola jaringan koneksi. Titik pada *breadboard* ada yang saling terhubung.

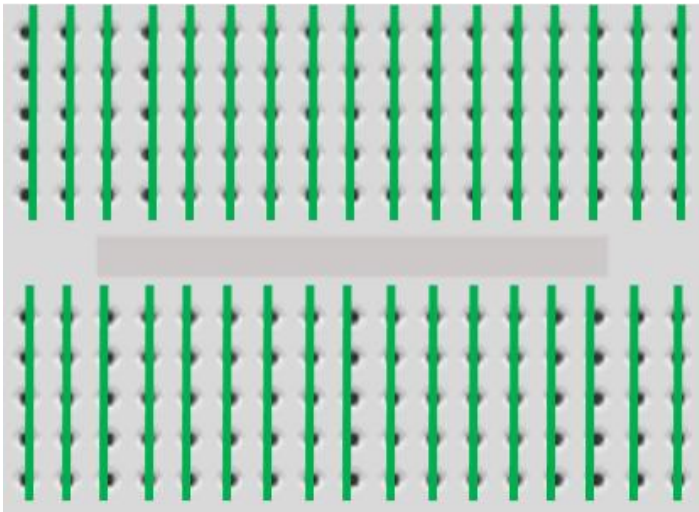
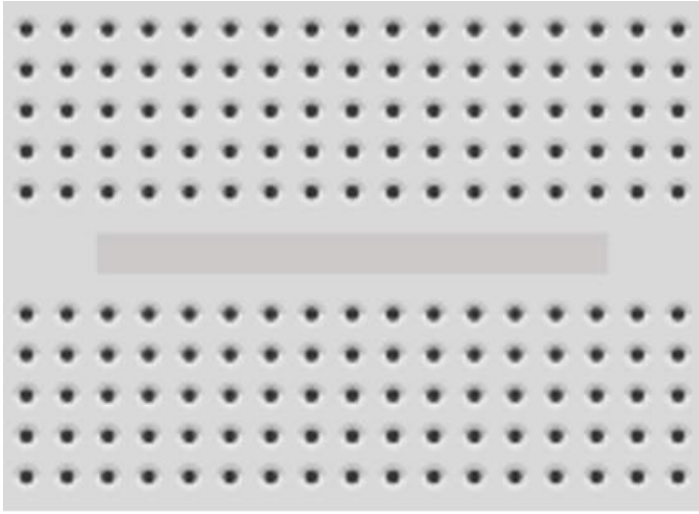
Terdapat beberapa ukuran *breadboard* yang ada di pasaran, yaitu mini, medium, dan besar. Gambar 1.5 menunjukkan pola koneksi pada *breadboard* berukuran besar. Lubang yang terhubung dengan garis pada gambar merupakan lubang-lubang yang terkoneksi sehingga lubang-lubang tersebut dapat dianggap sebagai titik yang sama. Gambar 1.6 dan 1.7 masing-masing menunjukkan pola koneksi pada *breadboard* berukuran medium dan mini.



Gambar 1.5 Pola koneksi pada *breadboard* berukuran besar.

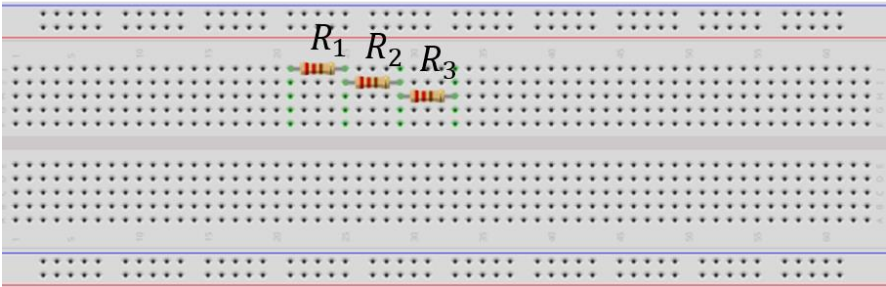
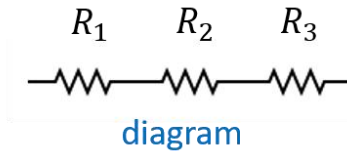


Gambar 1.6 Pola koneksi pada *breadboard* berukuran medium



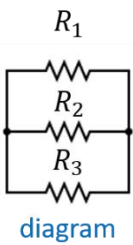
Gambar 1.7 menunjukkan pola koneksi pada *breadboard* berukuran mini

Berikut adalah contoh penggunaan *breadboard* untuk menyusun rangkaian



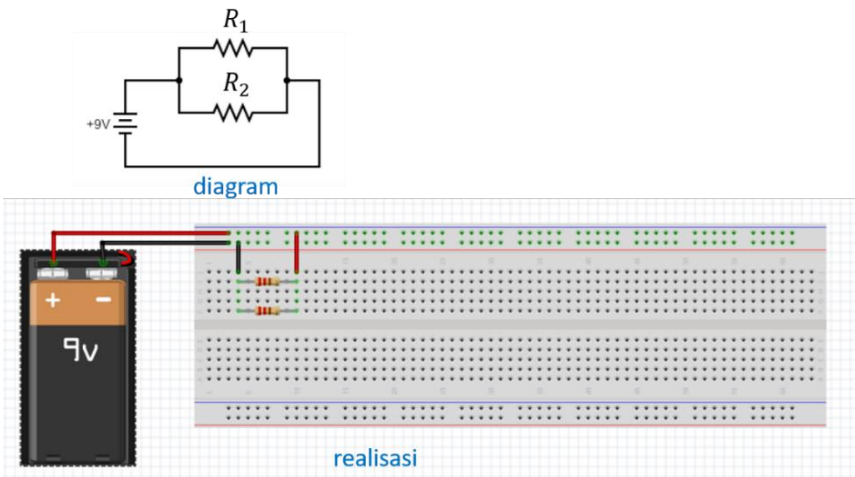
realisasi

Gambar 1.8 Contoh 1 realisasi rangkaian resistor seri yang disusun pada *breadboard*



realisasi




Gambar 1.9 Contoh 2 realisasi rangkaian resistor paralel yang disusun pada *breadboard*


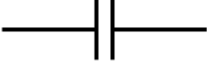
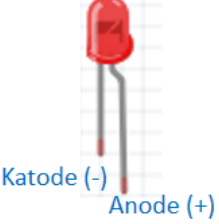


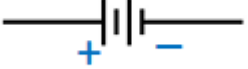






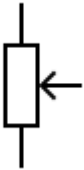






Gambar 1.9 Contoh 3 realisasi rangkaian yang disusun pada *breadboard*

Buku ini berfokus pada penggunaan Arduino Uno dalam percobaan fisika, khususnya materi kelistrikan. Pada rangkaian listrik, terdapat banyak komponen yang memiliki nama dan peran masing-masing. Saat menggambar diagram rangkaian, komponen-komponen listrik biasa digambarkan dengan symbol. Tabel 1.1 menunjukkan symbol beberapa komponen listrik pada diagram rangkaian.

Tabel 1.1 Simbol beberapa komponen listrik pada diagram rangkaian.

Komponen	Gambar	Simbol dalam Diagram
Resistor		 atau 

Komponen	Gambar	Simbol dalam Diagram
Kapasitor		
LED		
Baterai		
Dioda		
Potensioet er		 <p data-bbox="838 1512 897 1541">Atau</p>

Komponen	Gambar	Simbol dalam Diagram
		
Variable voltage supply		
Sumber arus AC		
Induktor	 <p data-bbox="409 1561 636 1599">Sumber: https://pxhere.com/id/photo/1248787</p>	

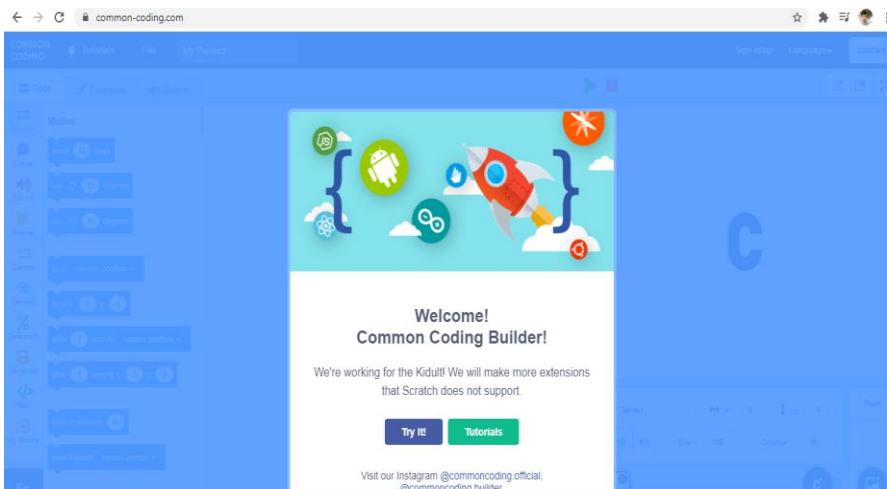
BAB 2

COMMON-CODING

Supaya dapat bekerja sesuai dengan yang diinginkan pengguna, Arduino perlu diberi perintah. Arduino dapat dihubungkan ke komputer melalui kabel USB kemudian diprogram. Ada beberapa software pemrograman Arduino. Kali ini akan dibahas pemrograman Arduino dengan *block-based language* yang ada dalam Common-Coding.

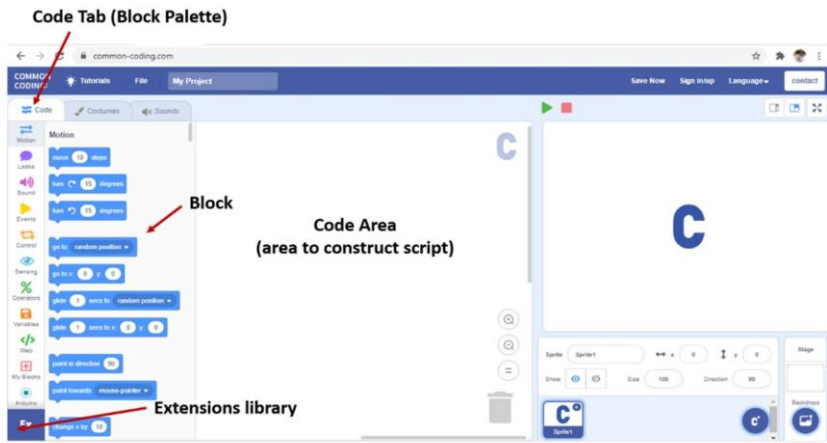
A. Pengenalan Common-Coding

Common-Coding merupakan bahasa pemrograman berbasis blok (*block-based programming*) yang tersedia online. Halaman Common-Coding dapat diakses melalui browser internet (misalnya Google Chrome), alamatnya adalah <https://Common-Coding.com/>. Tampilan awal setelah kita pergi ke halaman tersebut ditunjukkan seperti pada Gambar 2.1. Pilih *Try It* untuk masuk ke halaman utama.



Gambar 2.1 Tampilan awal Common-Coding

Halaman utama dapat disajikan pada Gambar 2.2 berikut



Gambar 2.2 Halaman utama Common-Coding

B. Menghubungkan Arduino ke Common Coding

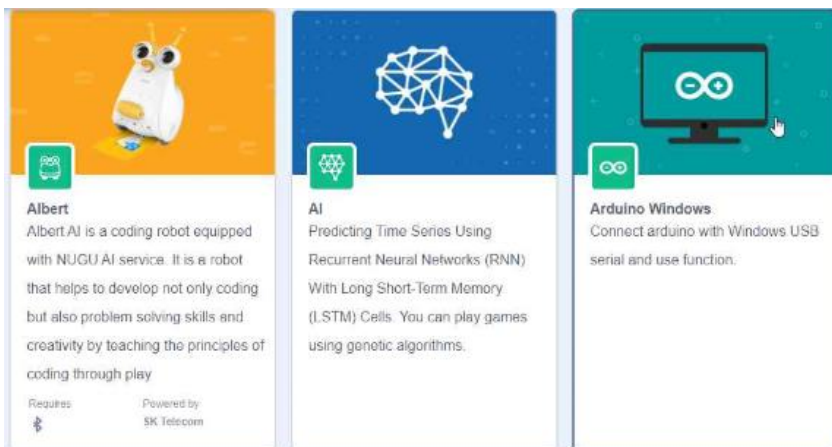
Common-Coding menyediakan beberapa fitur ekstensi, salah satunya adalah pemrograman dasar Arduino. Fitur inilah yang akan dimanfaatkan pada percobaan fisika. Untuk memulai pemrograman Arduino dengan Common-Coding, terdapat beberapa langkah:

1. Buka menu Ex (*extension*) pada pojok kiri bawah



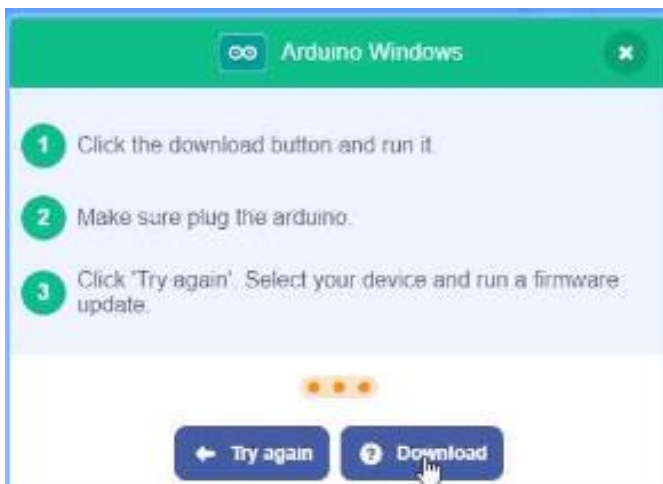
Gambar 2.3 Menu Ex (*Extension*) pada Common-Coding

2. Setelah diklik, akan muncul beberapa fitur extension. Pilih *Arduino for Windows*



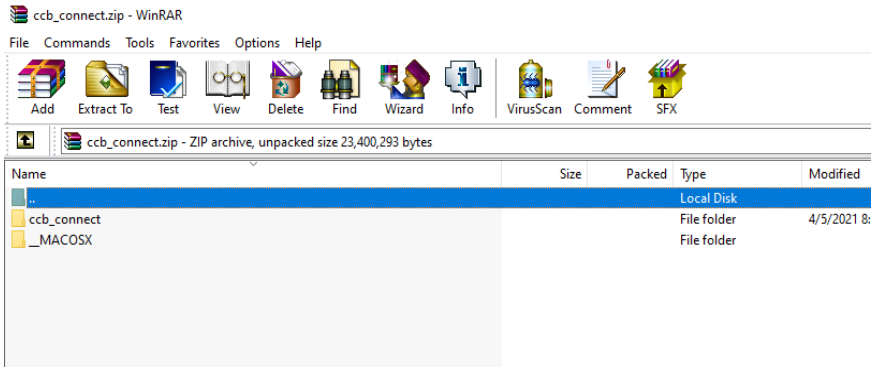
Gambar 2.4 Fitur *extension* pada Common-Coding

3. Berikutnya akan muncul tampilan kotak dialog *Arduino Windows*. Untuk menggunakan Arduino pada Common-Coding diperlukan aplikasi *framework*. Aplikasi ini bisa diperoleh dengan mengklik tombol *Download* pada kotak.



Gambar 2.3 Kotak dialog *Arduino Windows*

4. Bila tombol Download diklik, komputer akan mengunduh file bernama **ccb_connect.zip**. Kita perlu mengekstraknya terlebih dahulu.



Gambar 2.4 File pada folder ccb_connect.zip

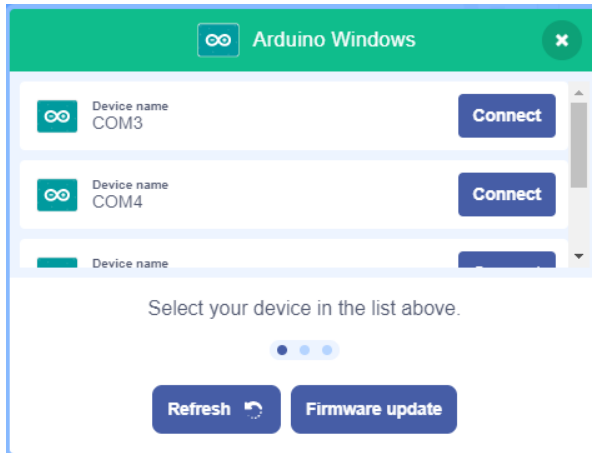
5. Di dalam file hasil ekstrak, terdapat folder bernama **ccb_connect** yang memuat file **ccb_connect.exe**. File tersebut merupakan aplikasi *frameware* untuk Windows yang harus dibuka setiap kita akan menggunakan Arduino pada Common-Coding. Bila menggunakan operasi sistem Mac, bisa menggunakan file yang ada pada **_MACOSX**. Program **ccb_connect.exe** membutuhkan Java untuk berjalan. Oleh karena itu, bila pada komputer belum ada Java, harus diinstall terlebih dahulu. Program Java dapat diunduh dari <https://www.java.com/en/download/>



Gambar 2.5 Tampilan ccb_connect.exe.

Aplikasi `ccb_connect.exe` tidak boleh ditutup selama kita menggunakan Arduino pada Common-Coding

- Selanjutnya, kita perlu kembali ke Common-Coding. Halaman `Common-Coding.com` di-*refresh*, kemudian klik kembali *Try It > Menu Ex > Arduino for Windows*. Setelah itu, akan muncul kotak dialog seperti pada Gambar 2.6.



Gambar 2.6 Tampilan Arduino Windows

- Pilih *Firmware Update*. Kemudian Klik Update pada COM dimana Arduino terhubung.



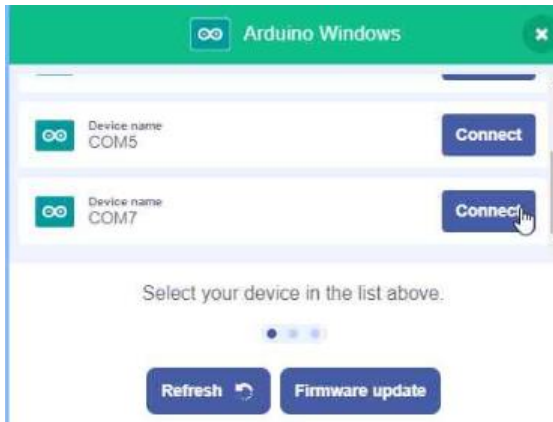
Gambar 2.7 Firmware update

8. Setelah proses selesai, klik OK



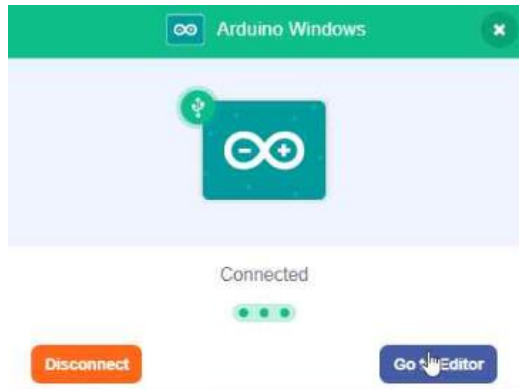
Gambar 2.8 Proses *Firmware update* telah selesai

9. Berikutnya akan muncul tampilan seperti pada gambar 2.9. Klik Connect pada COM7 dimana Arduino terhubung ke komputer.



Gambar 2.9 Menghubungkan COM 7

Bila proses koneksi selesai, akan muncul tampilan seperti pada Gambar 2.10. Klik *Go to Editor* untuk memulai menyusun program.

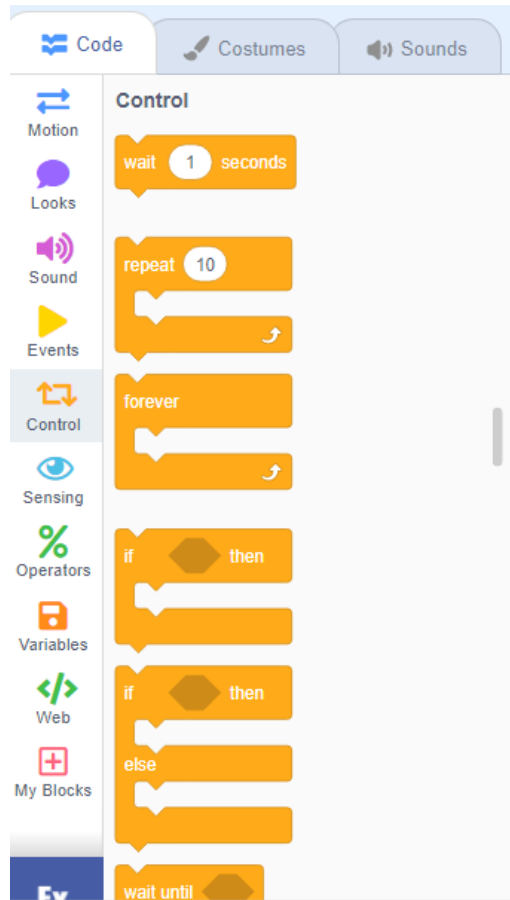


Gambar 2.10 COM 7 telah terkoneksi

Langkah 1-4 dilakukan hanya pada saat pertama kali menggunakan Common-Coding. Apabila akan menggunakan lagi berikutnya, langkah diawali dengan membuka aplikasi *ccb_connect.exe* kemudian membuka Common-Coding.com di *browser* dan melanjutkan langkah 5-9.

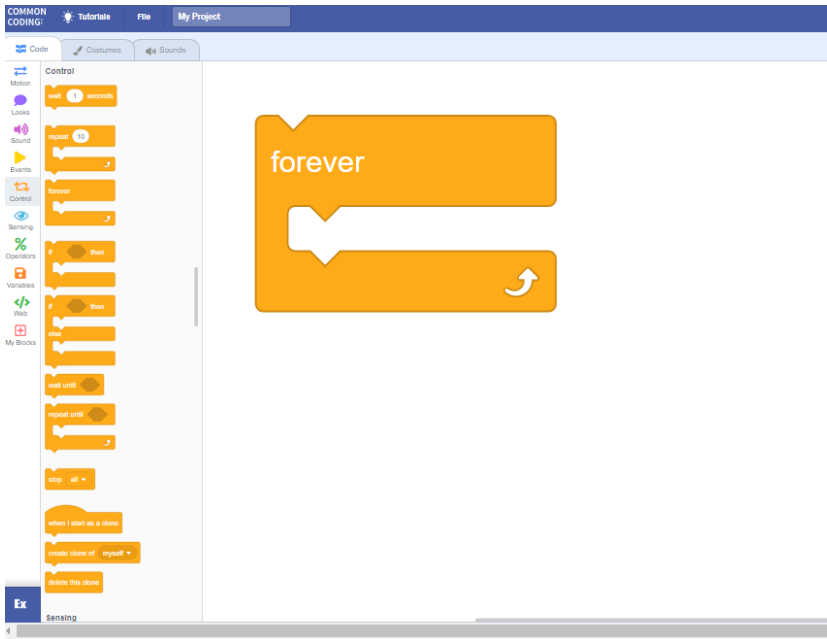
C. Langkah menyusun *script* pada Common-Coding

Common-Coding menyediakan bahasa pemrograman berbasis blok. *Script* program disusun dengan menyusun blok. Setiap blok memiliki perintahnya masing-masing. Pada panel kiri halaman Common-Coding, terdapat kumpulan *script* yang berbentuk blok, seperti ditunjukkan pada Gambar 2.11. Blok tersebut dikategorikan sebagai Motion, Looks Sound, Events, Control, Sensing, Operators, Variables, Web, Arduino, dll.



Gambar 2.11 Tampilan blok *script* pemrograman pada kategori Control

Jika kategori Control dipilih, akan muncul tampilan kumpulan blok seperti pada Gambar 2.11 di atas. Terdapat blok *wait*, *repeat*, *forever*, *if*, dll. Tiap blok memiliki kegunaan masing-masing. Bila kita ingin menggunakan salah satu blok untuk menyusun suatu program, caranya adalah klik salah satu blok kemudian *drag and drop* ke *code area*.



Gambar 2.12 Meletakkan blok *script* pada *code area* dengan cara *drag and drop*

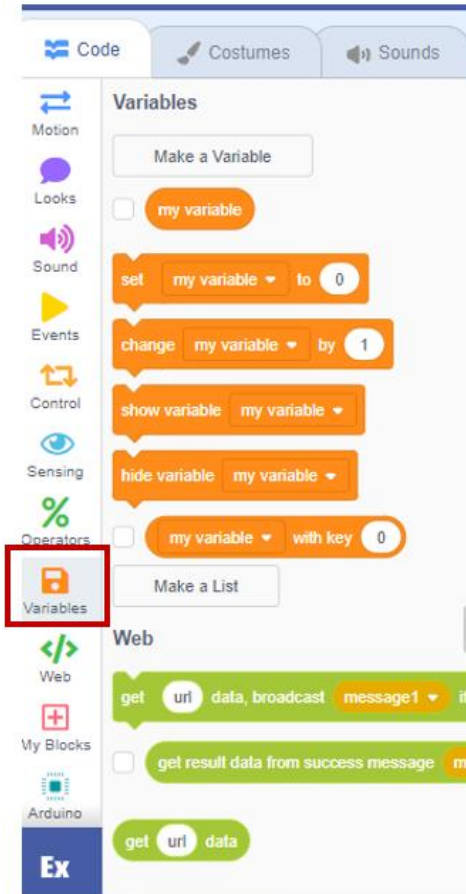
Sebuah program memerlukan gabungan beberapa blok, sehingga kita perlu memanggil blok lain dan menggabungkannya pada *code area*

- Posisi blok dapat diatur dengan *click and drag*.
- Blok dapat dihapus dengan klik kanan > *Delete Block*
- Blok dapat diduplikasi dengan cara klik kanan > *Duplicate*
- Catatan dapat diberikan pada setiap blok, dengan cara klik kanan > *Add comment*. Fitur ini bisa digunakan untuk memberikan pengingat tentang makna blok.

D. Langkah mendeklarasi variabel

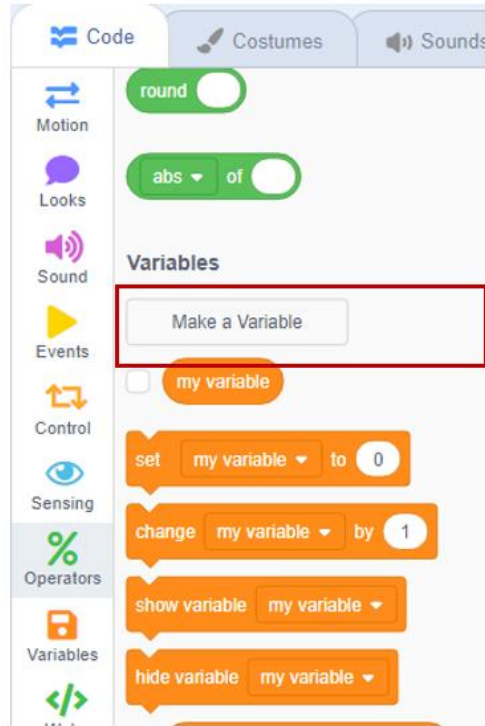
Pada saat membuat program dengan Common-Coding, terkadang kita perlu menggunakan variabel. Deklarasi variabel dapat dilakukan dengan:

1. Pilih menu variable pada bagian panel kiri



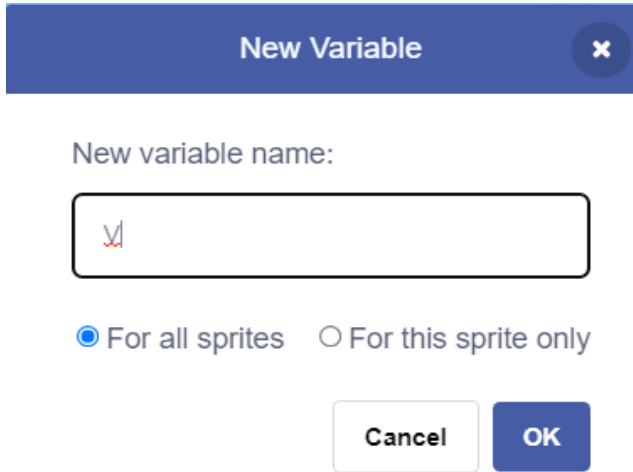
Gambar 2.13 Menu Variables

2. Untuk membuat atau menyatakan suatu variabel, pilih menu *Make a Variable*



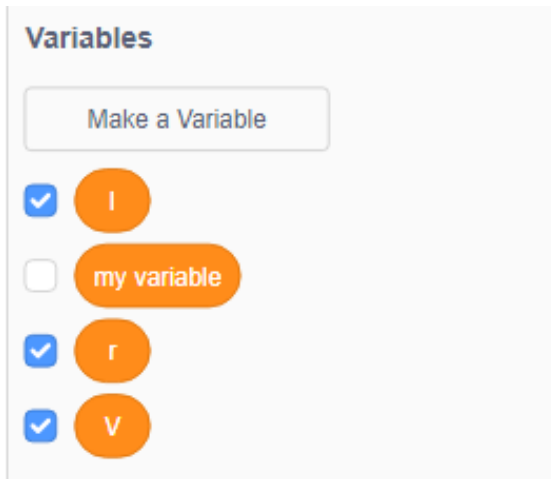
Gambar 2.14. Menu *Make a Variable* dipakai untuk membuat variable

3. Akan muncul kotak dialog *New Variable*. Pada *New variable name*, ketikkan nama variabel, misalnya 'V'. Lalu klik OK



Gambar 2.15. Memberikan nama variable

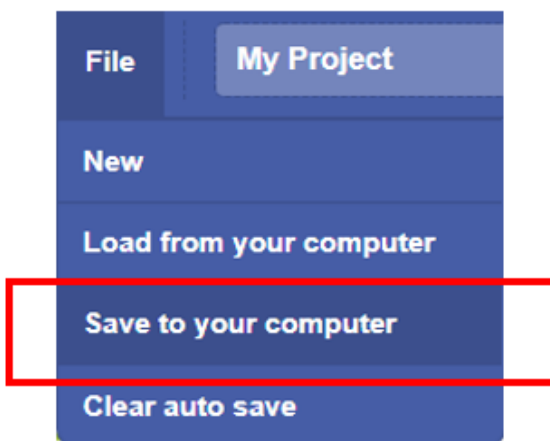
4. Ulangi langkah 2-3 untuk menambahkan variabel lainnya. Jika sudah ditambahkan, akan muncul daftar variabel seperti ditunjukkan pada Gambar 2.16.



Gambar 2.16. Tampilan daftar variabel yang telah dideklarasikan

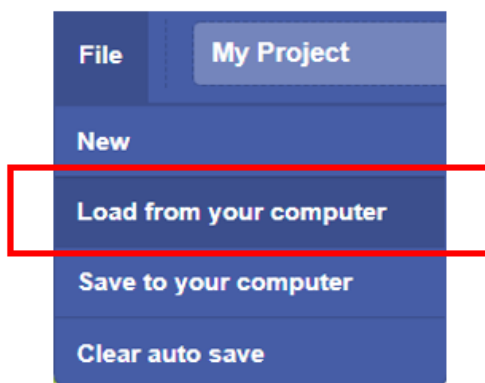
E. Menyimpan Proyek dan Membukanya Kembali

Proyek/program dapat disimpan pada komputer sehingga nantinya dapat digunakan kembali. Cara untuk menyimpan proyek adalah dengan memilih menu *File > Save to your computer*



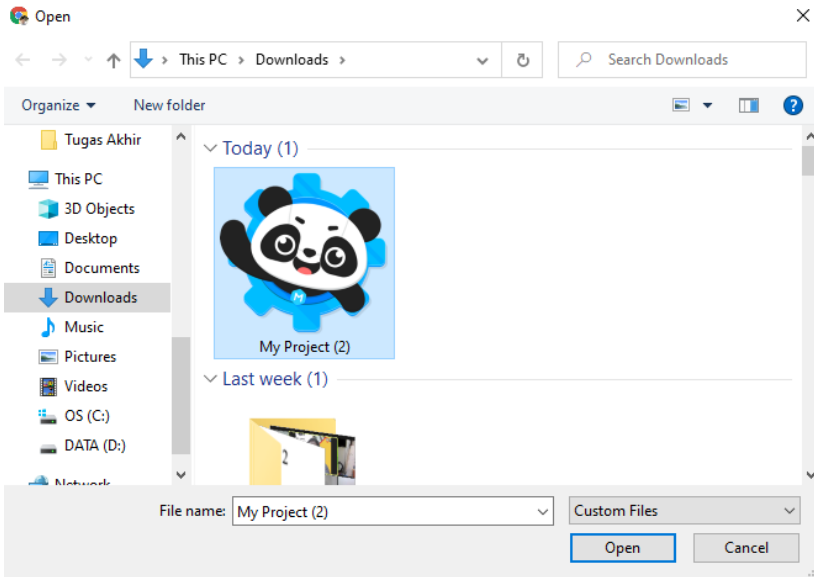
Gambar 2.17 Menyimpan proyek ke computer

Bila kita sudah memiliki proyek yang tersimpan sebelumnya, kita juga dapat mengedit atau menggunakannya dengan cara klik *File > Load from your computer*



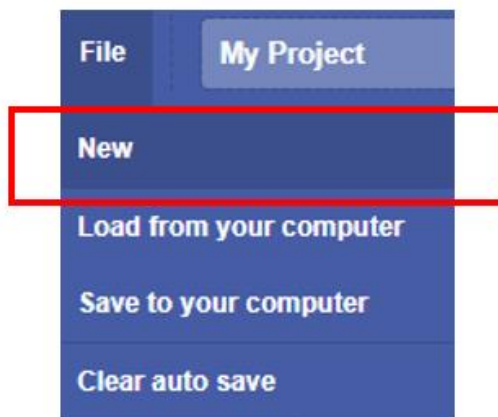
Gambar 2.18 Membuka proyek yang tersimpan pada komputer

Berikutnya akan muncul kotak dialog Open. Setelah kotak dialog Open terbuka, kita dapat memilih file proyek dan klik Open.



Gambar 2.19 Tampilan *Open Windows*

Setelah selesai membuat satu proyek, kita dapat membuat proyek baru, dengan cara pilih menu *File > New*.



Gambar 2.18 Membuat proyek baru

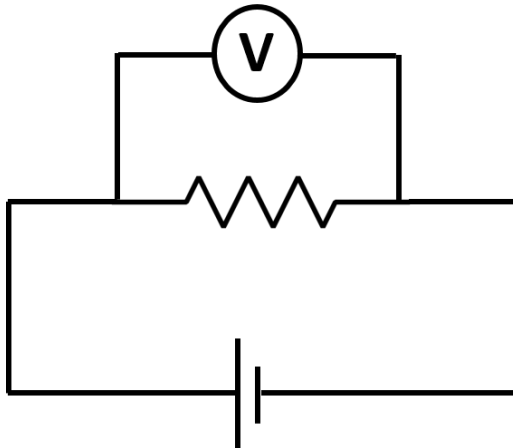
BAB 3

APLIKASI ARDUINO DALAM PERCOBAAN FISIKA

Bab ini akan membahas beberapa aplikasi Arduino Uno dalam praktikum fisika, khususnya pada materi kelistrikan. Arduino dapat dimanfaatkan sebagai pengganti alat ukur kelistrikan, seperti pengukur tegangan, arus, dan hambatan listrik. Selain itu, Arduino juga dapat dimanfaatkan untuk menyediakan sumber listrik DC (*direct current*) pada percobaan.

A. Pengukuran Tegangan

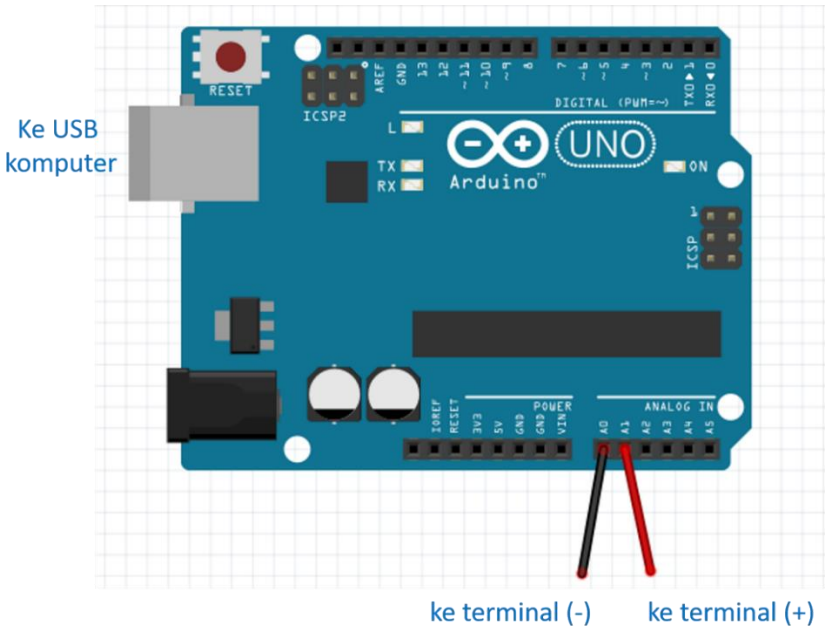
Tegangan dapat diukur dengan menggunakan voltmeter maupun sensor tegangan (*voltage sensor*). Untuk mengukur tegangan pada komponen rangkaian, voltmeter harus dipasang secara parallel dengan komponen yang ingin diukur tegangannya. Perhatikan contoh pada gambar 3.1, voltmeter dipasang secara parallel dengan hambatan R untuk mengukur tegangan pada hambatan R (V_R)



Gambar 3.1. Pemasangan voltmeter untuk mengukur tegangan pada resistor R

1. Pengukuran Tegangan dengan Menggunakan Arduino Uno

Arduino dapat dimanfaatkan untuk mengukur tegangan DC. Namun, besar tegangan maksimal yang dapat terbaca adalah 5V. Rangkaian yang digunakan untuk mengukur tegangan dapat dilihat pada Gambar 3.2.



Gambar 3.2 Rangkaian untuk pengukuran tegangan DC

Adapun persamaan yang dipakai untuk mendapatkan nilai tegangan adalah:

$$\text{Tegangan yang diukur} = 5 \times (A_1 - A_0) / 1023 \text{ (dalam Volt)} \quad (1)$$

Dimana,

A_1 = nilai pembacaan pin A_1

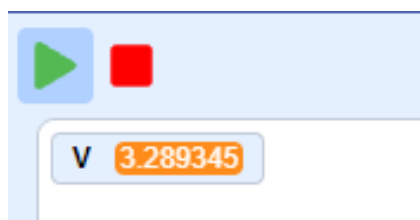
A_2 = nilai pembacaan pin A_0

Contoh implementasi persamaan (1) dalam *script* Common-Coding dapat dilihat pada Gambar 3.3. *Script* tersebut

memerintahkan Arduino untuk membaca tegangan setiap detiknya.



Gambar 3.3. Script untuk mengukur tegangan menggunakan Arduino



Gambar 3.4 Contoh tampilan pembacaan tegangan setelah script pada Gambar 3.3 dieksekusi.

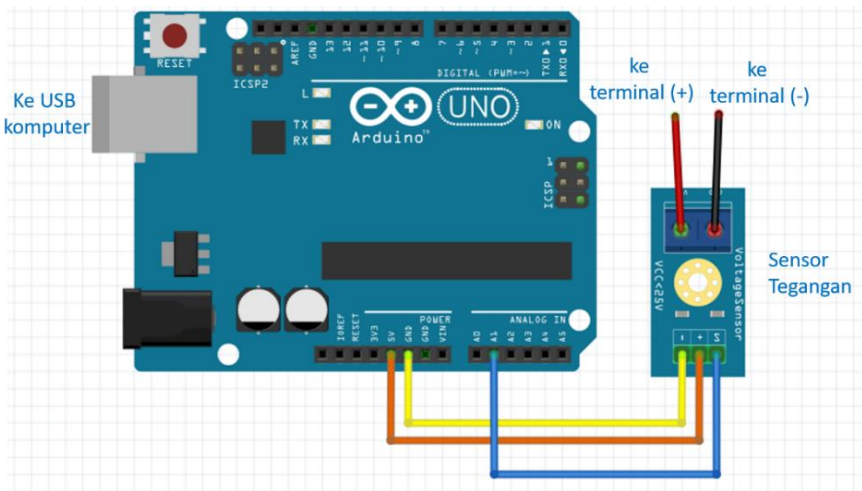
2. Pengukuran Tegangan dengan Menggunakan Sensor dan Arduino

Pengukuran tegangan dapat juga dilakukan dengan tambahan modul sensor tegangan. Penggunaan modul sensor tegangan pada Arduino memungkinkan pengukuran tegangan lebih dari 5 V. Masing-masing sensor memiliki ukuran maksimal tegangan yang dapat diukur, biasanya akan ada keterangan pada sensor tersebut, misalnya $VCC < 25V$ artinya batas maksimal yang dapat diukur adalah 25 V.

Seperti ditunjukkan pada Gambar 3.5, sensor tegangan memiliki dua pin pada terminal skrup (GND dan VCC) dan 3 pin

pada terminal *male* (S, -, +). Terminal skrup dihubungkan ke rangkaian yang akan diukur, sedangkan terminal *male* dihubungkan ke papan Arduino. Berikut adalah keterangan detailnya:

1. Pin 'S' (*sense*), dihubungkan ke Input Analog pada Arduino (misalnya pin A3)
2. Pin '+', dihubungkan ke pin 5V pada Arduino
3. Pin '-', dihubungkan ke GND pada papan Arduino
4. Pin 'GND' dihubungkan ke kutub negatif rangkaian yang akan diukur
5. Pin 'VCC' dihubungkan ke kutub positif rangkaian yang akan diukur



Gambar 3.5. Pemasangan sensor tegangan

Berikut adalah persamaan yang digunakan untuk mendapatkan nilai tegangan riil rangkaian yang diukur berdasarkan nilai sensor (*voltage sensor value*) (Dijk, 2019):

$$\text{Tegangan yang diukur} = 5,128 \times \text{Voltage Sensor Value} / 1023 \times 5 \quad (2)$$

Voltage sensor value merupakan nilai keluaran sensor yang masuk dan diolah oleh input analog Arduino. Jika rangkaian yang digunakan adalah seperti Gambar 3.5, maka dalam hal ini *voltage sensor value* adalah nilai pembacaan analog A1.

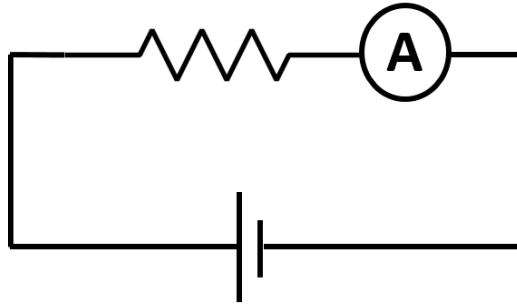
Pemrograman Arduino dapat menggunakan Common-Coding, dan Gambar 3.6 adalah contoh *script* untuk menampilkan hasil pengukuran tegangan tiap detik.



Gambar 3.6 Contoh *script* untuk mendapatkan hasil ukur tegangan menggunakan sensor

B. Pengukuran Arus

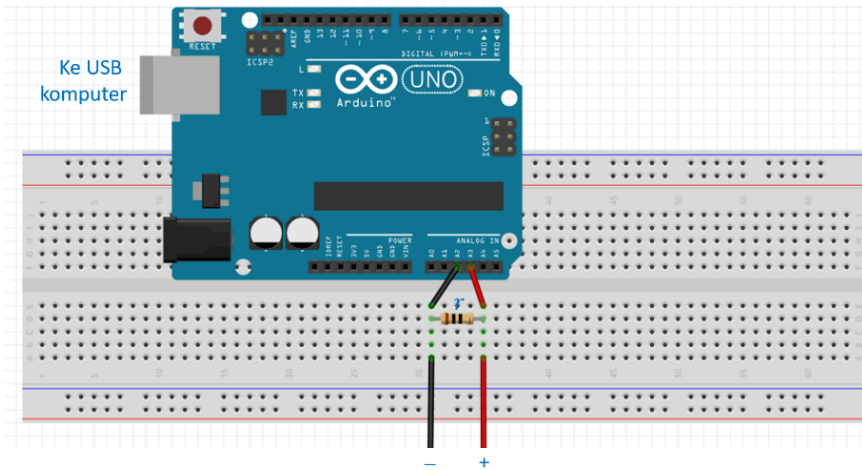
Arus listrik dapat diukur dengan menggunakan ammeter maupun sensor arus seperti ACS712. Untuk mengukur arus pada komponen rangkaian, pengukur arus harus dipasang secara seri dengan rangkaian. Perhatikan contoh pada Gambar 3.7. Ammeter dipasang secara seri dengan hambatan R untuk mengukur arus yang mengalir pada komponen R .



Gambar 3.7 Pemasangan ammeter untuk mengukur arus yang mengalir pada resistor R

1. Pengukuran Arus dengan Menggunakan Arduino

Rangkaian pengukur arus dapat dibuat dengan memanfaatkan papan Arduino Uno dan sebuah resistor. Gambar 3.8 menunjukkan rangkaian pengukur arus yang terdiri dari papan Arduino Uno, resistor dengan hambatan r , dan kabel penghubung.



Gambar 3.8 Rancang bangun pengukur arus dengan papan Arduino Uno dan Resistor $r = 10 \Omega$

Nilai hasil ukur arus menggunakan rangkaian pada Gambar 3.8 dapat diperoleh dengan persamaan:

$$V_r = (A_3 - A_2) \times 5/1023$$
$$I = \frac{V_r}{r} \times 1000 \quad (3)$$

Keterangan:

V_r = tegangan di r (dalam V)

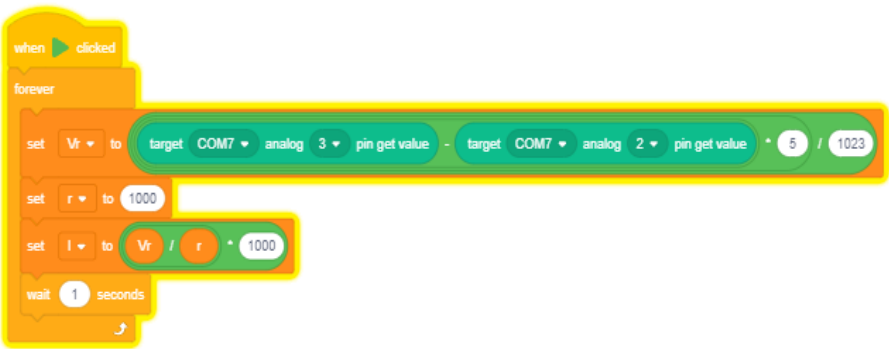
r = hambatan resistor yang terhubung pada Arduino Uno (misal 10 Ω)

A_2 = nilai pembacaan pin A2

A_3 = nilai pembacaan pin A3

I = arus yang diukur (dalam mA)

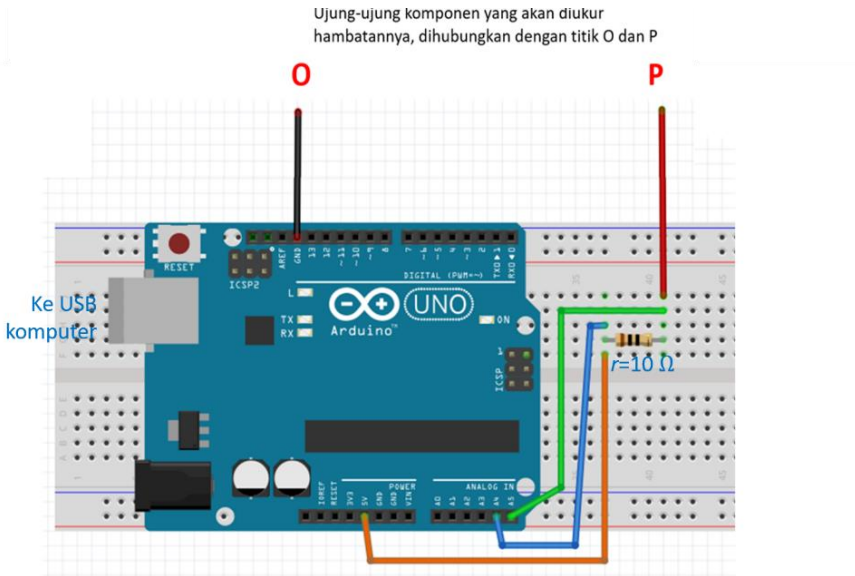
Contoh *script* implementasi persamaan (3) untuk menampilkan hasil pengukuran arus tiap detik dapat dilihat pada Gambar 3.9.



Gambar 3.9 Contoh *script* untuk mendapatkan hasil ukur arus dengan rangkaian pada Gambar 3.8

C. Pengukuran Hambatan

Arduino dapat digunakan untuk mengukur hambatan suatu bahan, misalnya hambatan konduktor. Gambar 3.10 menunjukkan contoh rangkaian yang dapat digunakan untuk mengukur hambatan.



Gambar 3.10 Rangkaian pengukur hambatan

Berikut adalah persamaan untuk mendapatkan nilai hambatan dari pengukuran dengan Arduino.

$$\begin{aligned} V_a &= \frac{A4}{1023} \times 5 \\ V_b &= \frac{A5}{1023} \times 5 \\ V_1 &= V_a - V_b \\ R &= (r \times (V_a - V_1)) / V_1 \end{aligned} \tag{4}$$

Keterangan:

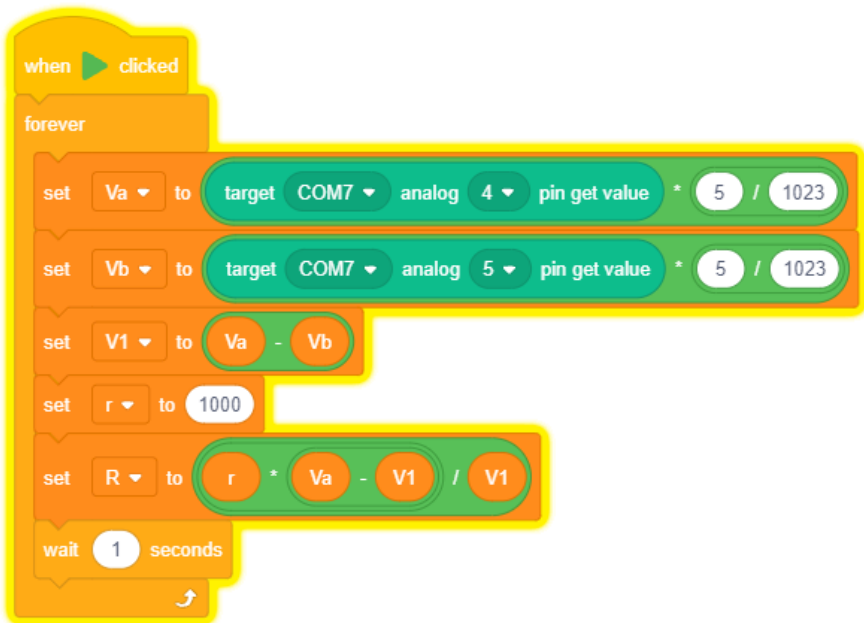
R =hambatan yang diukur (dalam Ω)

r = hambatan resistor yang terhubung pada Arduino Uno (100 Ω)

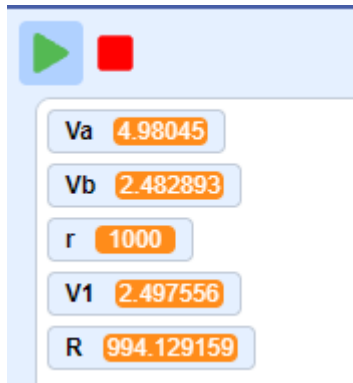
$A4$ = nilai pembacaan potensial pin A4

$A5$ = nilai pembacaan potensial pin A5

Contoh *script* membaca hambatan dengan rangkaian di atas ditunjukkan pada Gambar 3.11.



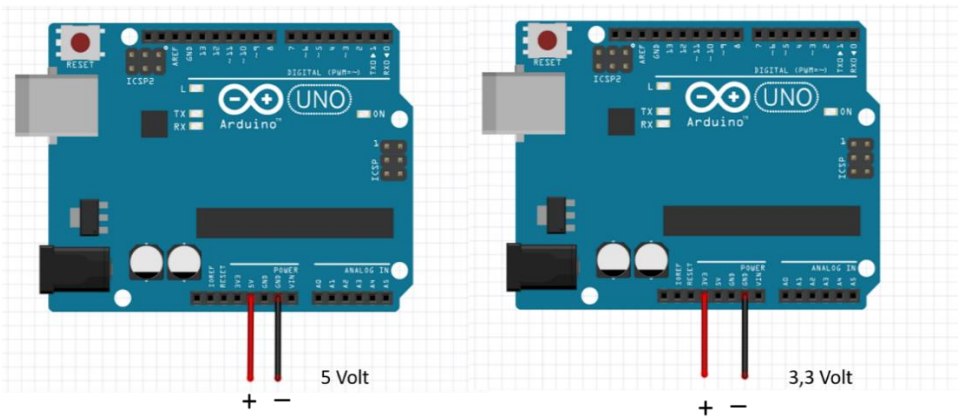
Gambar 3.11 Contoh *script* untuk membaca hambatan komponen



Gambar 3.12 Contoh hasil pembacaan hambatan

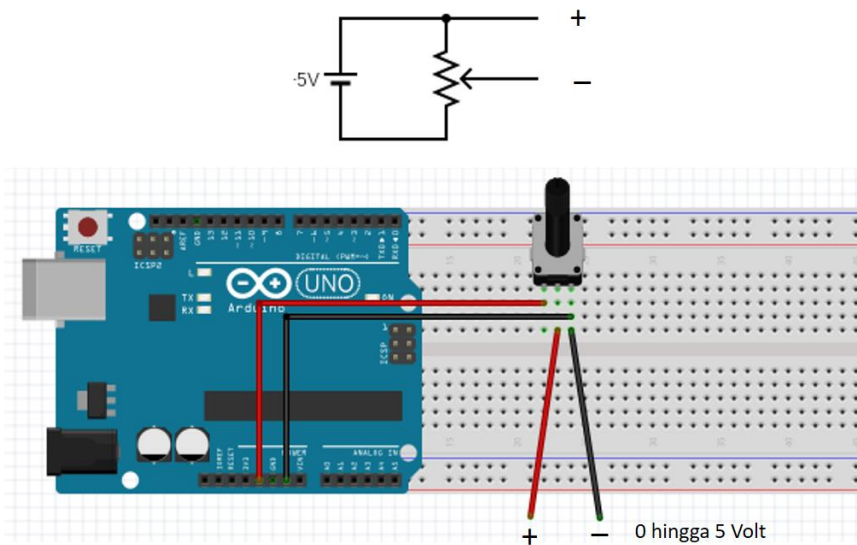
D. Menggunakan Arduino Sebagai Sumber Tegangan DC

Arduino Uno memiliki pin sumber daya 5V dan 3,3 V. Tanpa harus diprogram, pin tersebut dapat digunakan seperti baterai dengan tegangan 5 dan 3,3 V.



Gambar 3.13 Menggunakan Arduino Uno sebagai sumber tegangan DC 5 volt (kiri) dan 3,3 volt (kanan)

Apabila menginginkan tegangan yang dapat diatur dari 0 hingga 5 V, rangkaian di atas dapat dikombinasikan dengan rangkaian pembagi tegangan menggunakan potensiometer.

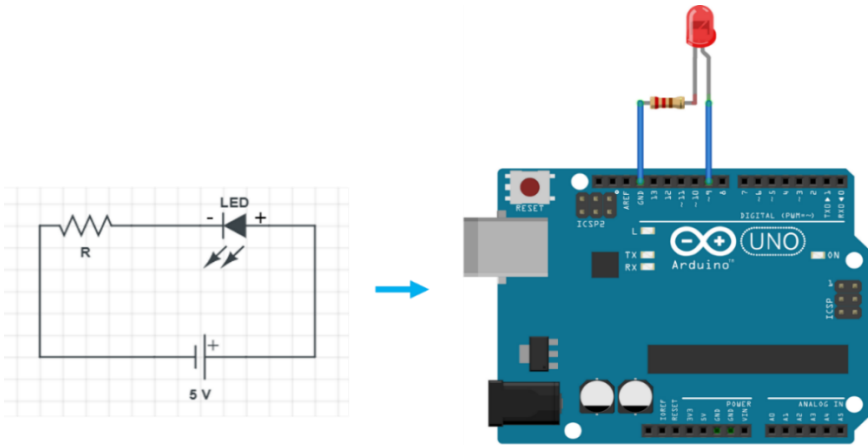


Gambar 3.14 Rangkaian pembagi tegangan dihubungkan ke pin 5 V dapat menghasilkan tegangan antara 0 hingga 5 V.

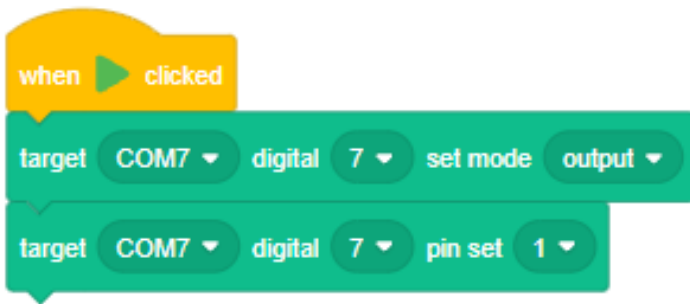
Digital pin pada Arduino Uno juga dapat digunakan sebagai mode output atau mode *DigitalWrite*. Hanya ada 2 pilihan luaran/output pin digital yaitu HIGH dan LOW.

- Jika diset HIGH (atau 1), maka pin akan mengeluarkan tegangan sekitar 5V
- Jika diset LOW (atau 0), maka pin akan mengeluarkan tegangan 0 V, atau sama saja sumber tegangan menjadi OFF.

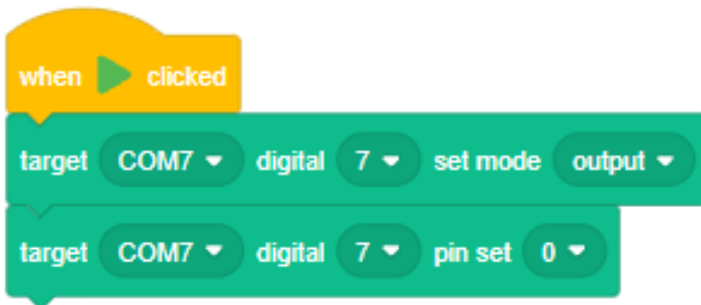
Contoh rangkaian dengan sumber tegangan 5V dari pin digital Arduino Uno dapat dilihat pada Gambar 3.15. Jenis keluaran pin digital perlu diprogram. Gambar 3.16 dan 3.17 merupakan contoh *script* Common-Coding untuk mengatur pin digital dalam mode output HIGH dan LOW.



Gambar 3.15. Contoh rangkaian dengan sumber tegangan 5V dari pin digital Arduino Uno



Gambar 3.16 *Script* untuk mengatur pin digital 7 mengeluarkan tegangan HIGH (1) atau V=5 volt



Gambar 3.17 *Script* untuk mengatur pin digital 9 mengeluarkan tegangan LOW (0) atau V=0 volt

BAB 4

CONTOH PERCOBAAN FISIKA MENGGUNAKAN ARDUINO

Pada Bab 3, telah dibahas pemanfaatan Arduino Uno sebagai alat ukur besaran pada materi kelistrikan dan sumber tegangan DC berikut dengan perograman menggunakan Common-Coding. Pada bab 4 ini, aplikasi tersebut akan dikombinasikan untuk melakukan beberapa percobaan fisika pada materi kelistrikan. Contoh percobaan yang dibahas antara lain percobaan Hukum Ohm, pengukuran hambatan pengganti, menginvestigasi *ohmic* dan *non-ohmic material*, Hukum Kirchhoff, dan rangkaian resistor-kapasitor (RC).

A. Percobaan Hukum Ohm (Hubungan $V-I$ pada Konduktor)

Tujuan

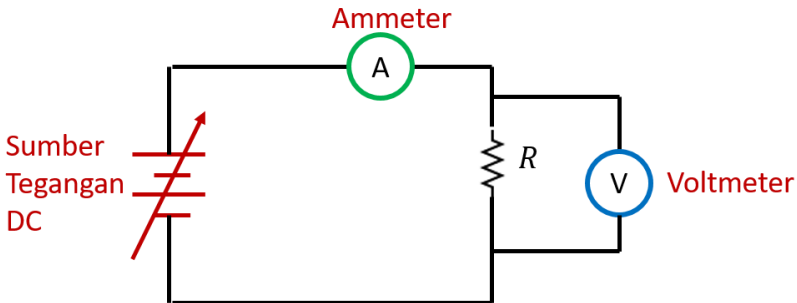
1. Mengetahui hubungan antara tegangan dan arus pada konduktor
2. Menghitung hambatan berdasarkan pengukuran tegangan dan arus

Bahan

1. Resistor 10 Ohm (1 buah)
2. Resistor 100, 200, 300 Ohm (masing-masing 1 buah)
3. Kabel jumper male-male (9 buah)
4. Potensiometer $1k\Omega$ (1 buah)
5. Arduino UNO (1 buah)
6. Kabel serial USB (1 buah)
7. *Breadboard* (1 buah)

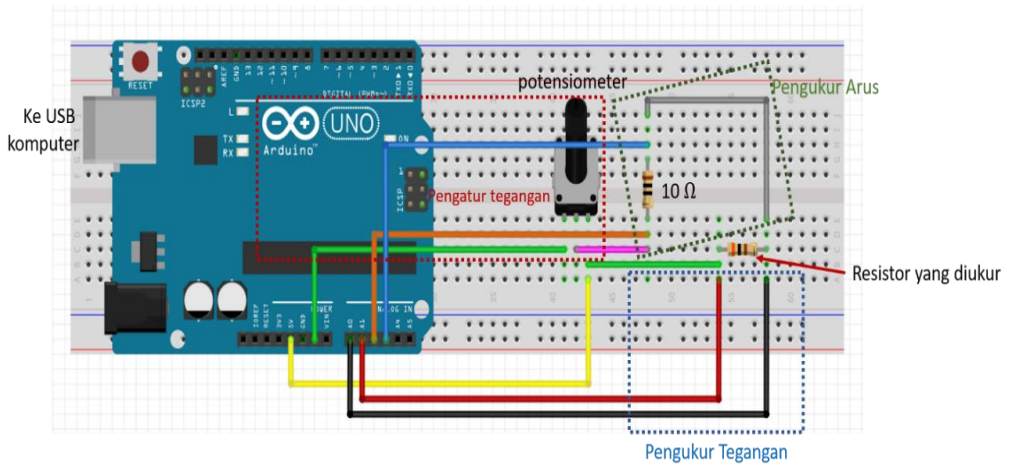
Langkah Percobaan

1. Menyiapkan bagian rangkaian sumber tegangan DC dengan output yang dapat diatur antara 0-5 volt menggunakan Arduino, lihat Gambar 3.14
2. Menyiapkan bagian rangkaian pengukuran arus (ammeter) dengan Arduino, lihat Gambar 3.8
3. Menyiapkan bagian rangkaian pengukuran tegangan (voltmeter) dengan Arduino, lihat Gambar 3.2
4. Merangkai rangkaian sebagai berikut dengan menggunakan bagian rangkaian pada langkah 1-3.



Gambar 4.1 Diagram rangkaian percobaan Hukum Ohm

5. Mengatur tegangan dengan mengatur potensiometer pada sumber tegangan. Setelah itu, tegangan dan arus pada resistor R diukur.
6. Mengubah tegangan hingga 5 variasi dan mengukur arus yang mengalir pada R.
7. Mencatat kode warna resistor
8. Mengulangi langkah 5-7 untuk resistor yang berbeda.



Gambar 4.2 Contoh rangkaian riil yang digunakan untuk percobaan Hukum Ohm



Gambar 4.3 Contoh *script* yang digunakan untuk pengambilan data menggunakan rangkaian Gambar 4.2

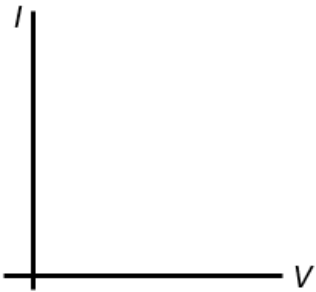
Tabulasi Data Percobaan

1. Resistor 1

Kode warna:

R (berdasarkan kode warna) = ... \pm ... Ω

V (volt)	I
0,5	
1,0	
1,5	
2,0	
2,5	
3,0	

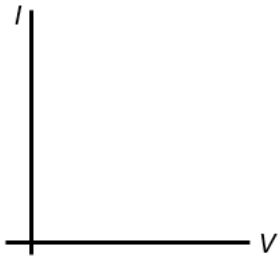


2. Resistor 2

Kode warna:

R (berdasarkan kode warna) = ... \pm ... Ω

V (volt)	I
0,5	
1,0	
1,5	
2,0	
2,5	
3,0	



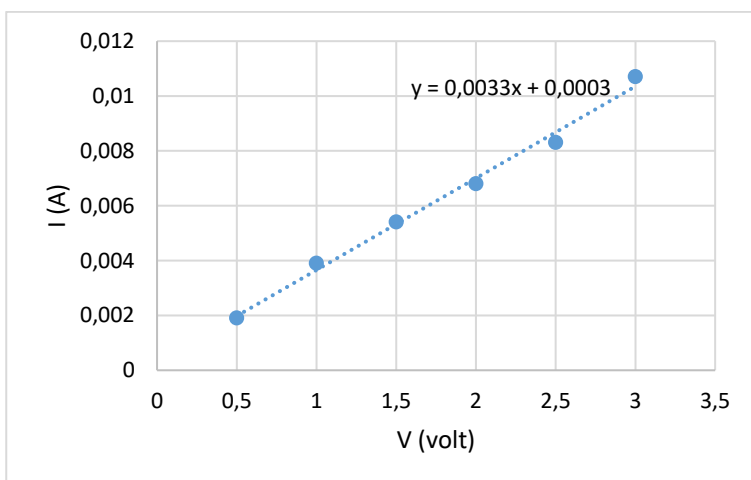
Contoh Hasil Percobaan

Resistor 1

Kode warna: orange-hitam-coklat-emas

R (berdasarkan kode warna) = $300 \pm 5\% \Omega$

V (volt)	I (A)
0,5	0,0019
1	0,0039
1,5	0,0054
2	0,0068
2,5	0,0083
3	0,0107



Gambar 4.4 Hubungan V - I berdasarkan data percobaan

Berdasarkan grafik hasil percobaan untuk Resistor 1, diperoleh kesimpulan bahwa I berbanding lurus dengan V .

$$I \propto V$$

$$I = \frac{1}{R} V$$

Gradien grafik di atas adalah 0,0033; sehingga pada percobaan ini, $\frac{1}{R} = 0,0033$ atau $R = 303 \Omega$. Hasil ini mendekati nilai yang tercantum pada kode warna resistor.

Kesimpulan yang Ingin Diperoleh

Percobaan ini mengarahkan peserta didik untuk menemukan konsep:

1. Pada konduktor, arus berbanding lurus dengan tegangan
2. Nilai hambatan dapat dihitung dari gradient grafik I - V

Model matematis yang dapat diperoleh:

$$I \propto V$$
$$I = \frac{1}{R} V$$

B. Percobaan Hukum Ohm (Mengidentifikasi Faktor Penentu Hambatan pada Konduktor)

Tujuan

1. Menginvestigasi pengaruh panjang konduktor pada hambatan
2. Menginvestigasi pengaruh luas penampang konduktor pada hambatan

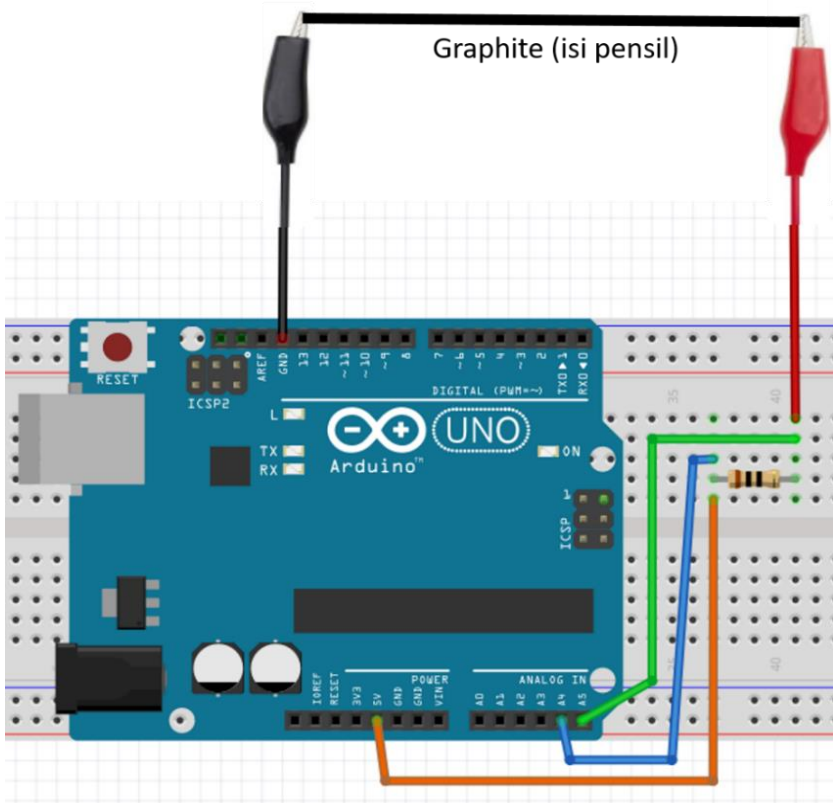
Bahan

1. Grafit/isi pensil 2B berdiameter 0,3, 0,5, 0,7, dan 0,9 mm dari merk yang sama dengan panjang yang sama (masing-masing 1 buah)
2. Grafit/isi pensil ukuran 2,0 mm, panjang: 9 cm (1 buah)
3. Penggaris
4. Papan Arduino (1 buah)
5. Kabel jumper *male-male* (5 buah)
6. Kabel USB serial (1 buah)
7. *Breadboard* (1 buah)
8. Kabel buaya (2 buah)
9. Resistor 10 Ohm (1 buah)

Langkah Percobaan

Percobaan 1

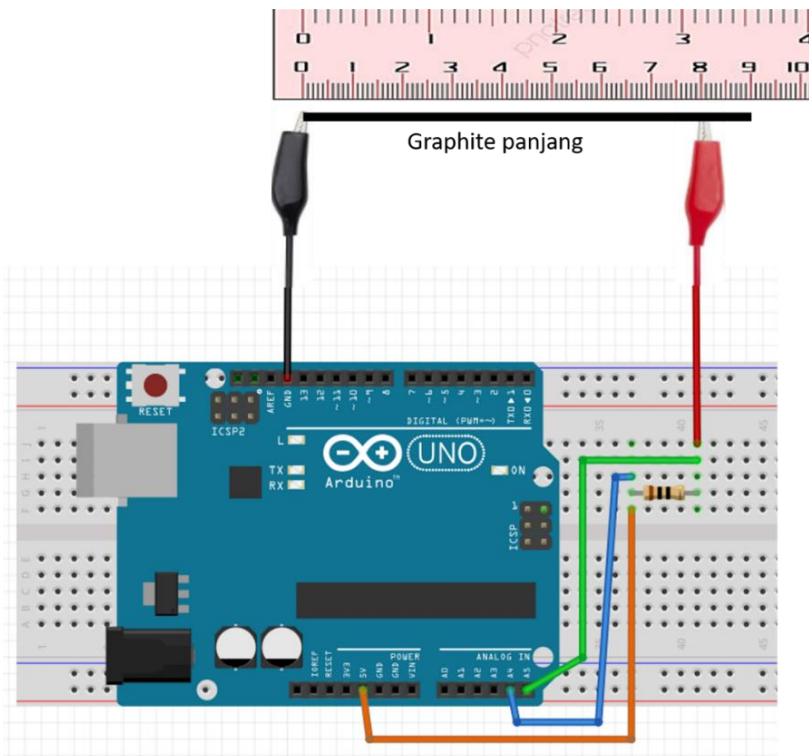
1. Menyiapkan rangkaian pengukur hambatan listrik dengan menggunakan Arduino (Lihat Gambar 3.11)
2. Mengukur panjang grafit/isi pensil dengan diameter 0,3, 0,5, 0,7, dan 0,9 mm
3. Mengukur hambatan grafit/isi pensil dengan berbagai diameter (0,3, 0,5, 0,7, dan 0,9 mm) dan panjang yang sama. Untuk melakukan pengukuran, ujung-ujung grafit dapat dicapit dengan kabel buaya, seperti ditunjukkan pada Gambar 4.5.



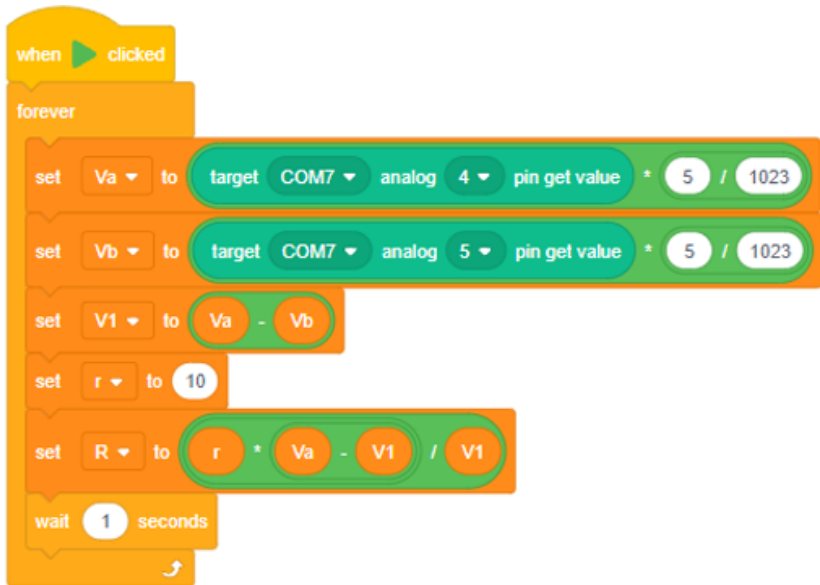
Gambar 4.5 Rangkaian percobaan untuk mengukur hambatan grafit

Percobaan 2

1. Mengambil grafit yang agak panjang
2. Salah satu ujung grafit tersebut dicapitkan ke salah satu kabel buaya pada pengukur hambatan. Kemudian kabel buaya yang lain dicapitkan di titik berjarak 3 cm dari ujung grafit tadi. Selanjutnya, lakukan pengukuran hambatan.
3. Memindahkan posisi kabel buaya ke titik yang jaraknya lebih jauh dan mengukur hambatannya.
4. Mengulangi langkah 4 hingga 6 variasi panjang.



Gambar 4.6 Rangkaian percobaan untuk mengukur hambatan grafit dengan variasi panjang

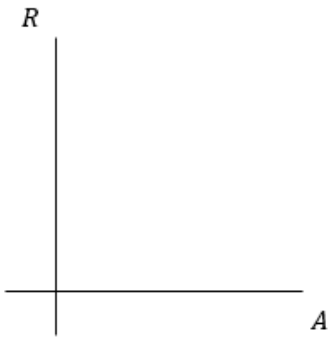


Gambar 4.7 Contoh *script* yang digunakan untuk pengambilan data menggunakan rangkaian Gambar 4.5 dan 4.6

Tabulasi Data Percobaan

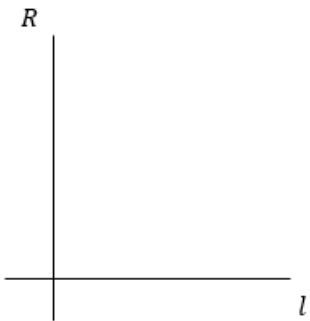
Percobaan 1

Panjang (l)	Diameter (d)	Luas penampang $A_{penampang} = \pi \left(\frac{d}{2}\right)^2$	R
	0,3 mm		
	0,5 mm		
	0,7 mm		
	0,9 mm		



Percobaan 2

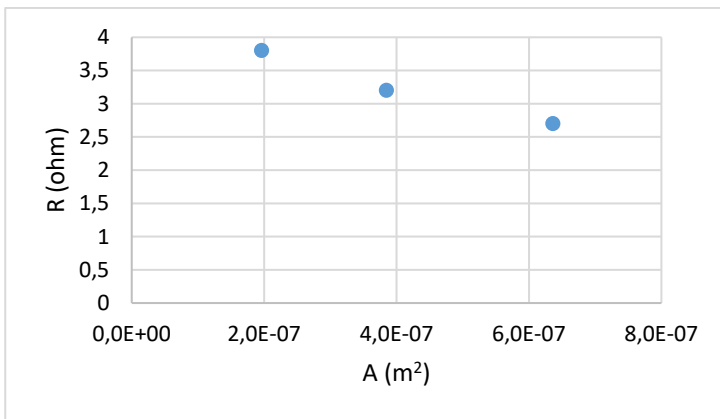
Diameter (d)	Panjang (l)	R
2,0 mm	3 cm	
2,0 mm	4 cm	
2,0 mm	5 cm	
2,0 mm	6 cm	
2,0 mm	7 cm	
2,0 mm	8 cm	



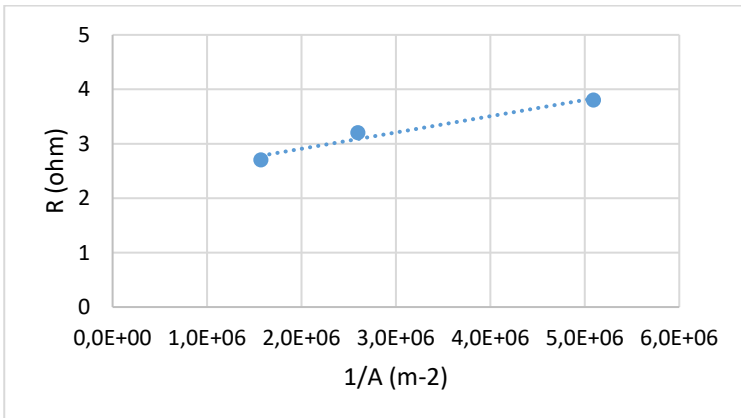
Contoh Hasil Percobaan

Percobaan 1

l (m)	d (m)	A (m ²)	$1/A$ (m ⁻²)	R (ohm)
0,09	0,0005	1,96E-07	5092958	3,8
0,09	0,0007	3,85E-07	2598448	3,2
0,09	0,0009	6,36E-07	1571901	2,7



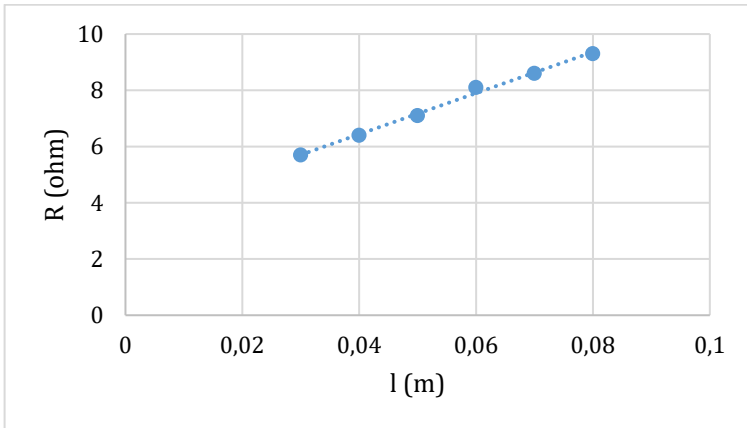
Gambar 4.8 Grafik hubungan hambatan (R) dan luas penampang konduktor (A) berdasarkan hasil percobaan



Gambar 4.9 Grafik hubungan hambatan (R) dan $1/A$ berdasarkan hasil percobaan

Percobaan 2

d (mm)	l (m)	R (ohm)
0,002	0,03	5,7
0,002	0,04	6,4
0,002	0,05	7,1
0,002	0,06	8,1
0,002	0,07	8,6
0,002	0,08	9,3



Gambar 4.10 Grafik hubungan hambatan (R) dan panjang konduktor (l) berdasarkan hasil percobaan

Kesimpulan yang Ingin Diperoleh

Percobaan ini mengarahkan peserta didik untuk menemukan konsep:

1. Hambatan konduktor berbanding lurus dengan panjang konduktor

2. Hambatan konduktor berbanding terbalik dengan luas penampang konduktor

Model matematis yang dapat diperoleh: $R \propto \frac{l}{A}$

C. Percobaan Mengidentifikasi *Ohmic* dan *Non-Ohmic Material*

Tujuan

Mengetahui hubungan antara tegangan (V) dan arus (I) pada berbagai bahan

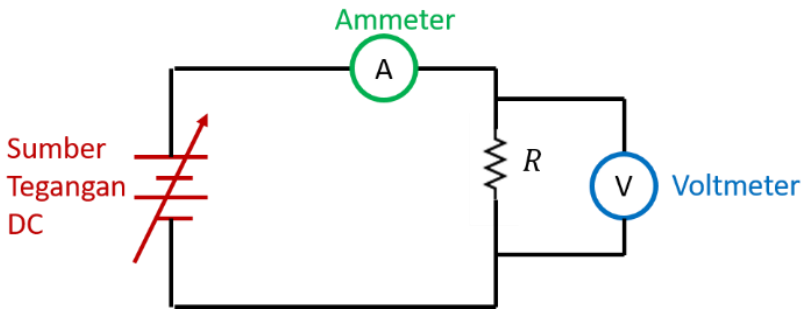
Bahan

1. Resistor 10 Ohm (1 buah)
2. Resistor 300 Ohm (1 buah)
3. *Light emitting diode* (LED) (1 buah)
4. Lampu filamen (1 buah)
5. Kabel *jumper male-male* (9 buah)
6. Potensiometer $1k\Omega$ (1 buah)
7. Arduino UNO (1 buah)
8. Kabel serial USB (1 buah)
9. *Breadboard* (1 buah)

Langkah Percobaan

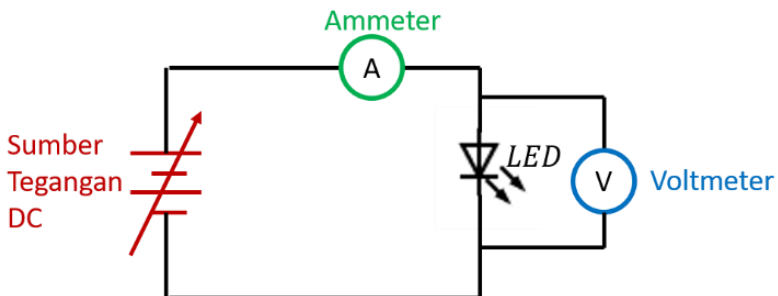
1. Menyiapkan bagian rangkaian sumber dan pengatur tegangan DC dengan Arduino (Lihat Gambar 3.14)
2. Menyiapkan bagian rangkaian pengukuran arus (ammeter) dengan Arduino, (Lihat Gambar 3.8)

3. Menyiapkan bagian rangkaian pengukuran tegangan (voltmeter) dengan Arduino, (Lihat Gambar 3.2)
4. Merangkai rangkaian sebagai berikut dengan menggunakan bagian rangkaian pada langkah 1-3.



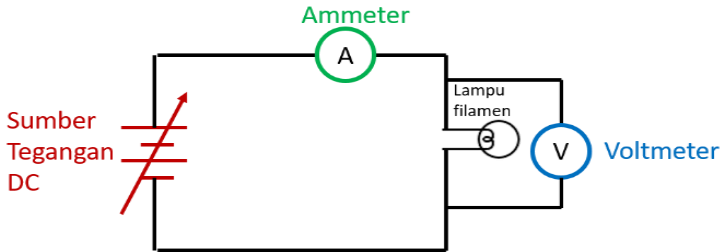
Gambar 4.11 Diagram rangkaian pengukuran V dan I pada resistor

5. Mengatur tegangan dengan mengatur potensiometer pada sumber tegangan. Setelah itu, tegangan dan arus pada resistor R diukur.
6. Mengubah tegangan hingga 6 variasi dan mengukur arus yang mengalir pada R .
7. Mengganti resistor dengan LED, kemudian mengulangi langkah 5-6

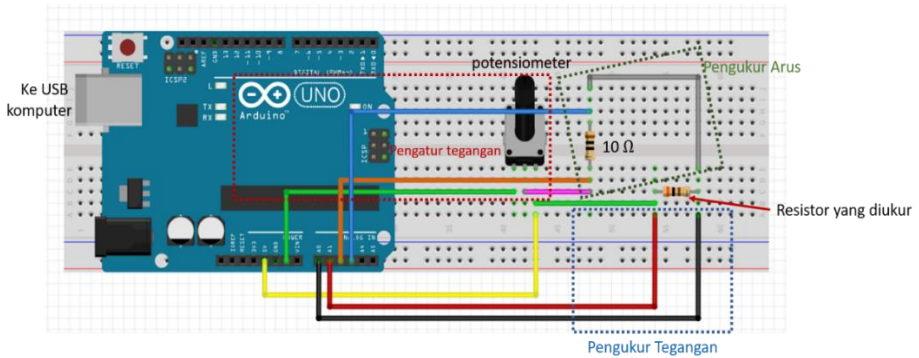


Gambar 4.12 Diagram rangkaian pengukuran V dan I pada LED

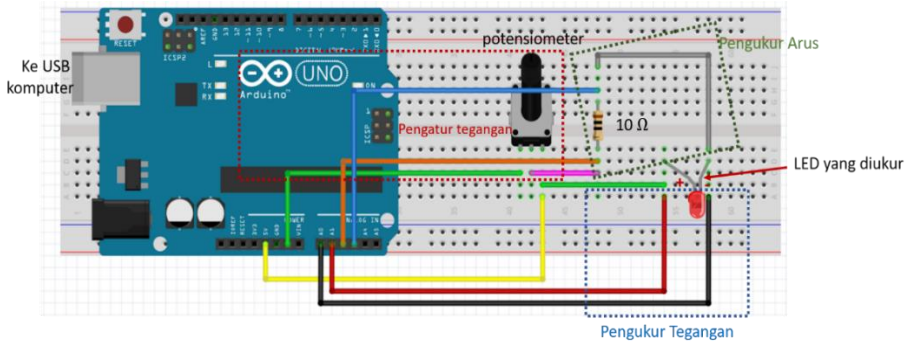
8. Mengganti resistor dengan lampu filamen, kemudian mengulangi langkah 5-6



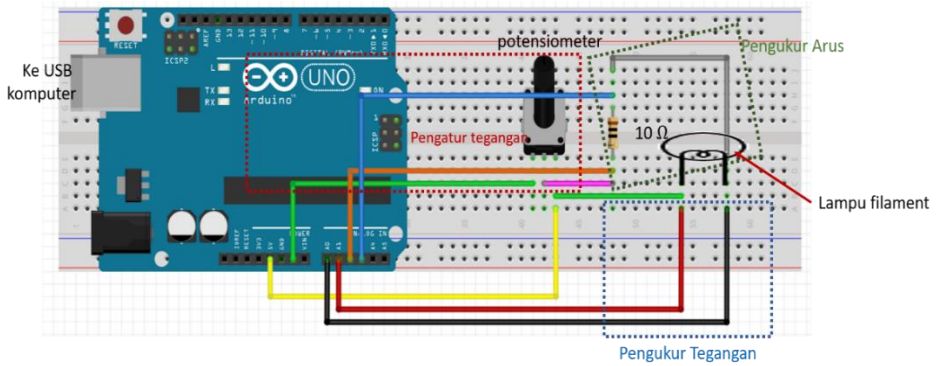
Gambar 4.13 Diagram rangkaian pengukuran V dan I pada lampu filamen



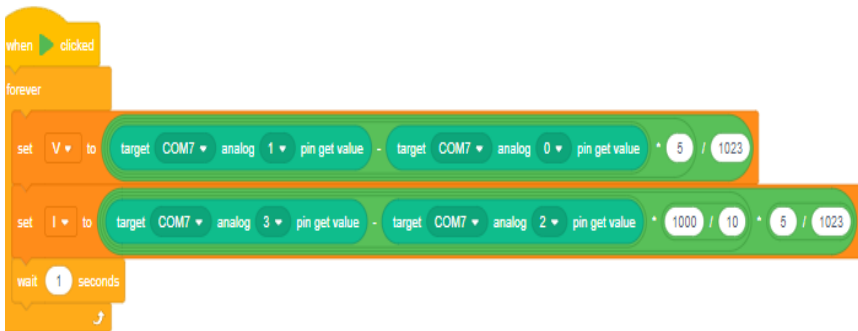
Gambar 4.14 Contoh realisasi diagram rangkaian pada Gambar 4.11



Gambar 4.15 Contoh realisasi diagram rangkaian pada Gambar 4.12



Gambar 4.15 Contoh realisasi diagram rangkaian pada Gambar 4.13



Gambar 4.16 Contoh *script* yang digunakan untuk pengambilan data V dan I menggunakan rangkaian Gambar 4.13, 4.14, dan 4.15

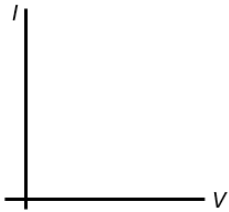
Tabulasi Data Percobaan

1. Resistor

Kode warna:

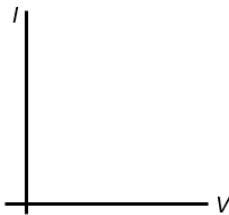
R (berdasarkan kode warna) = ... \pm ... Ω

V (volt)	I
0,5	
1,0	
1,5	
2,0	
2,5	
3,0	



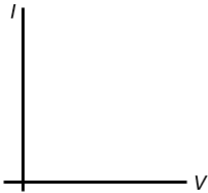
2. LED

V (volt)	I
0,5	
1,0	
1,5	
2,0	
2,5	
3,0	



3. Lampu Filamen

V (volt)	I
0,5	
1,0	
1,5	
2,0	
2,5	
3,0	



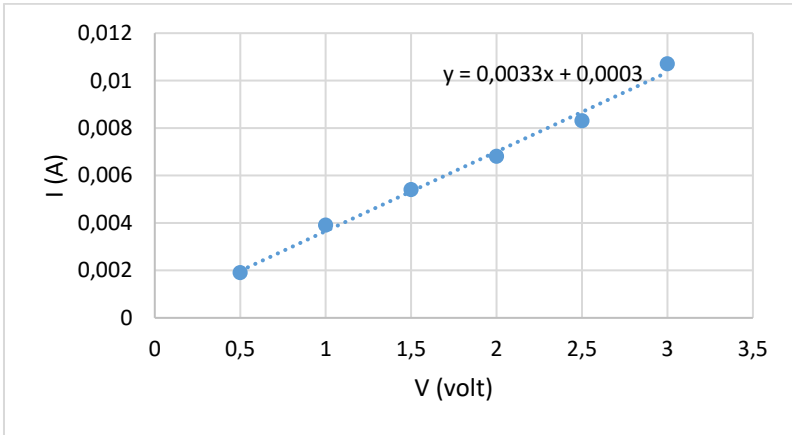
Contoh Hasil Percobaan

1. Resistor

Kode warna: orange-hitam-coklat-emas

R (berdasarkan kode warna) = $300 \pm 5\% \Omega$

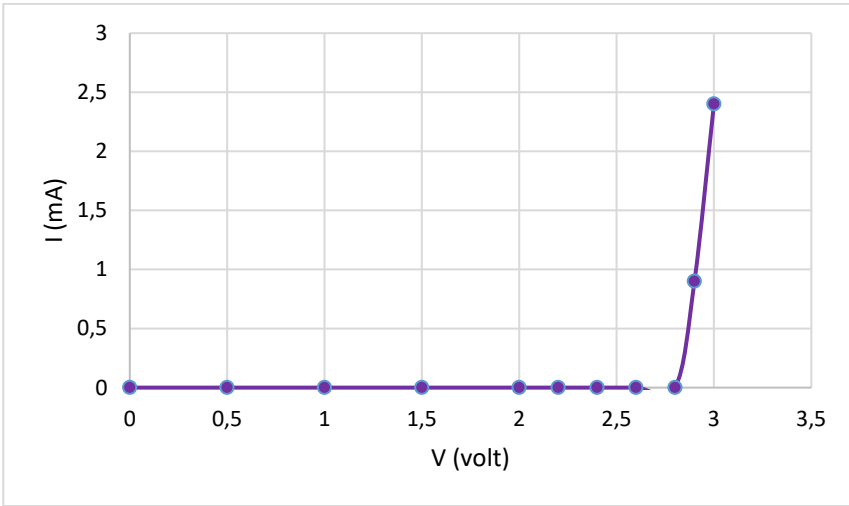
V (volt)	I (A)
0,5	0,0019
1	0,0039
1,5	0,0054
2	0,0068
2,5	0,0083
3	0,0107



Gambar 4.17 Hubungan V-I pada resistor berdasarkan data percobaan

2. LED

V (volt)	I (A)
0	0
0,5	0
1	0
1,5	0
2	0
2,2	0
2,4	0
2,6	0
2,8	0
2,9	0,9
3	2,4



Gambar 4.18 Hubungan V-I pada LED berdasarkan data percobaan

Kesimpulan yang Ingin Diperoleh

Percobaan ini mengarahkan peserta didik untuk menemukan konsep:

1. Hubungan $V-I$ tidak selalu linier pada semua jenis bahan
2. Bahan yang memiliki $V-I$ linier atau mengikuti hukum Ohm disebut ohmic material, contoh: konduktor (termasuk resistor)
3. Bahan yang tidak mengikuti hukum Ohm disebut non-ohmic material, contoh: LED yang terbuat dari semikonduktor.

D. Percobaan Menentukan Hambatan Pengganti Seri dan Paralel

Tujuan

1. Memodelkan hambatan pengganti resistor seri
2. Memodelkan hambatan pengganti resistor paralel

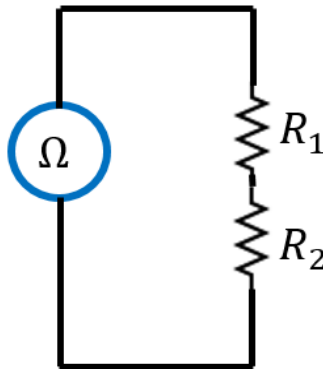
Bahan

1. Resistor 100, 200, 300 Ohm (masing-masing 1)
2. Kabel jumper male-male (5 buah)
3. *Breadboard* (1 buah)
4. Resistor 10 Ohm (1 buah)
5. Arduino (1 buah)
6. Kabel Serial USB (1 buah)

Langkah Percobaan

Percobaan 1

1. Menyiapkan rangkaian pengukur hambatan listrik dengan menggunakan Arduino (Lihat Gambar 3.11)
2. Merangkai 2 atau lebih resistor secara seri



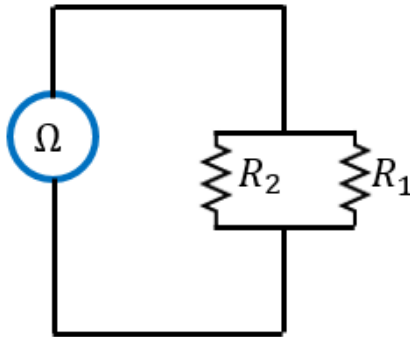
Gambar 4.19 Diagram rangkaian resistor secara seri dan pengukuran hambatannya

3. Mengukur hambatan rangkaian resistor seri tersebut menggunakan pengukur hambatan listrik yang telah disusun

4. Mengukur hambatan masing-masing resistor
5. Membuat rangkaian resistor seri dengan kombinasi R yang berbeda, lalu mengulangi langkah 3-4

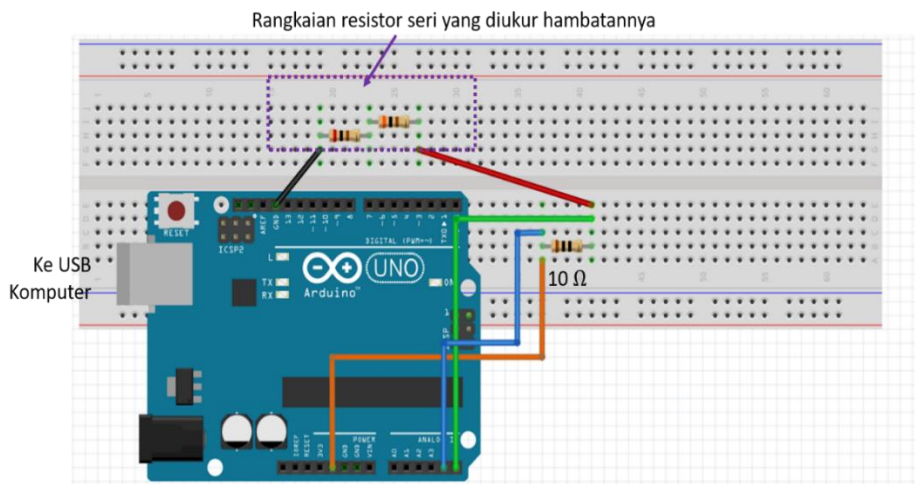
Percobaan 2

1. Menyiapkan rangkaian pengukur hambatan listrik dengan menggunakan Arduino (Lihat Gambar 3.11)
2. Merangkai 2 atau lebih resistor secara paralel
3. Mengukur hambatan rangkaian resistor paralel tersebut menggunakan pengukur hambatan listrik yang telah disusun

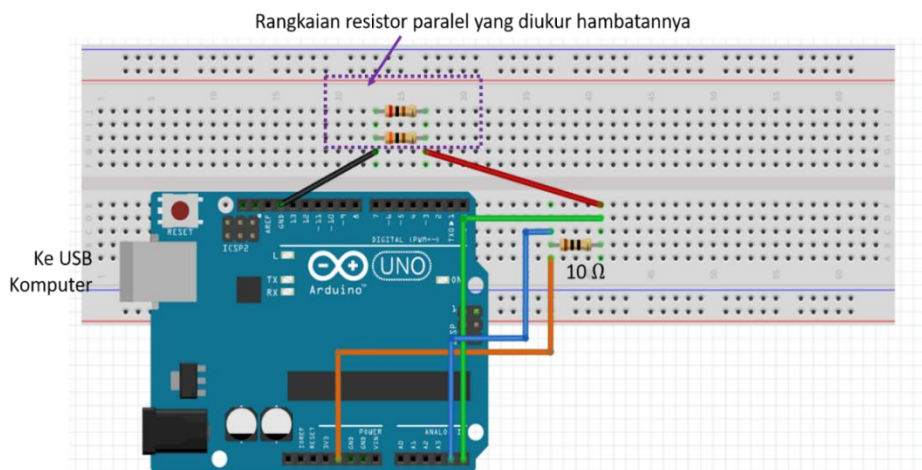


Gambar 4.19 Diagram rangkaian resistor secara paralel dan pengukur hambatannya

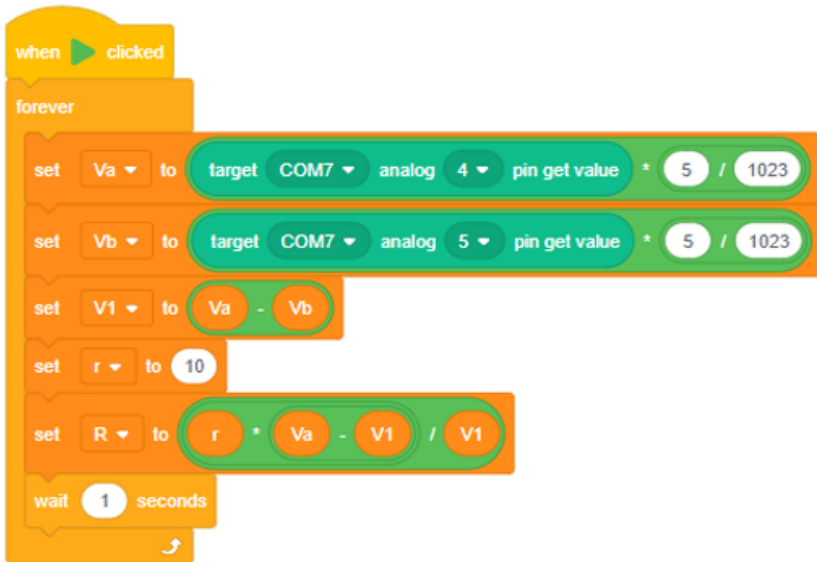
4. Mengukur hambatan masing-masing resistor
5. Membuat rangkaian resistor paralel dengan kombinasi R yang berbeda, lalu mengulangi langkah 3-4



Gambar 4.20 Contoh rangkaian untuk mengukur hambatan resistor yang dirangkai secara seri



Gambar 4.21 Contoh rangkaian untuk mengukur hambatan resistor yang dirangkai secara paralel



Gambar 4.21 Contoh *script* untuk mengambil data percobaan tentang hambatan menggunakan rangkaian di Gambar 4.19 dan 4.20

Tabulasi Data Percobaan

Percobaan 1

Rangkaian Seri	R_1		R_2		R_3		R_s
	Kode warna	Pengukuran	Kode warna	Pengukuran	Kode warna	Pengukuran	
1	100 Ω	...	200 Ω	...	-
2	100 Ω	...	300 Ω	...	-
3	100 Ω	...	200 Ω	...	300 Ω

Percobaan 2

Rangkaian Seri	R_1		R_2		R_3		R_p
	Kode warna	Pengukuran	Kode warna	Pengukuran	Kode warna	Pengukuran	
1	100 Ω	...	200 Ω	...	-
2	100 Ω	...	300 Ω	...	-
3	100 Ω	...	200 Ω	...	300 Ω

Contoh Hasil Percobaan

Data Percobaan 1

Rangkaian Seri	R_1		R_2		R_3		R_s
	Kode warna	Pengukuran	Kode warna	Pengukuran	Kode warna	Pengukuran	
1	100 Ω	99 Ω	200 Ω	203 Ω	-	-	300 Ω
2	100 Ω	99 Ω	300 Ω	300 Ω	-	-	399 Ω
3	100 Ω	99 Ω	200 Ω	203 Ω	300 Ω	300 Ω	592 Ω

Analisis Data Percobaan 1

R_1 (Ω)	R_2 (Ω)	R_3 (Ω)	R_s (Ω)	$R_1 + R_2 + \dots$ (Ω)
99	203	-	300	302
99	300	-	399	399
99	203	300	592	602

Data Percobaan 2

Rangkaian Paralel	$R_1 (\Omega)$		$R_2 (\Omega)$		$R_3 (\Omega)$		$R_s (\Omega)$
	Kode warna	Pengukuran	Kode warna	Pengukuran	Kode warna	Pengukuran	
1	100	99	200	203	-	-	66
2	100	99	300	300	-	-	73
3	100	99	200	203	300	300	55

Analisis Data Percobaan 2

$\frac{1}{R_1} (\Omega^{-1})$	$\frac{1}{R_2} (\Omega^{-1})$	$\frac{1}{R_3} (\Omega^{-1})$	$\frac{1}{R_p} (\Omega^{-1})$	$\frac{1}{R_1} + \frac{1}{R_2} + \dots (\Omega^{-1})$
0,010101	0,004926	-	0,015152	0,015027
0,010101	0,003333	-	0,013699	0,013434
0,010101	0,004926	0,003333	0,018182	0,01836

Kesimpulan yang Ingin Diperoleh

Percobaan ini mengarahkan peserta didik untuk menemukan konsep:

1. Hubungan hambatan pengganti pada rangkaian seri mengikuti $R_s = R_1 + R_2 + \dots$
2. Hubungan hambatan pengganti pada rangkaian seri mengikuti $\frac{1}{R_p} = \frac{1}{R_1} + \frac{1}{R_2} + \dots$

E. Percobaan Hukum Kirchhoff

Tujuan

1. Memodelkan jumlah tegangan pada *loop*
2. Memodelkan arus mengalir pada titik percabangan dalam suatu rangkaian listrik

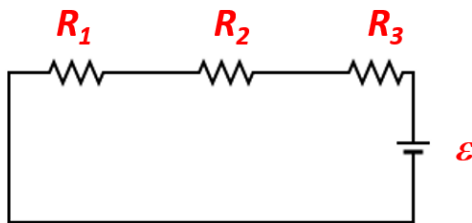
Bahan

1. Resistor 100, 200, 300 Ω (masing-masing 1)
2. Kabel *jumper male-male* (9 buah)
3. *Breadboard* (1 buah)
4. Resistor 10 Ohm (1 buah)
5. Arduino (1 buah)
6. Kabel Serial USB (1 buah)

Langkah Percobaan

Percobaan 1

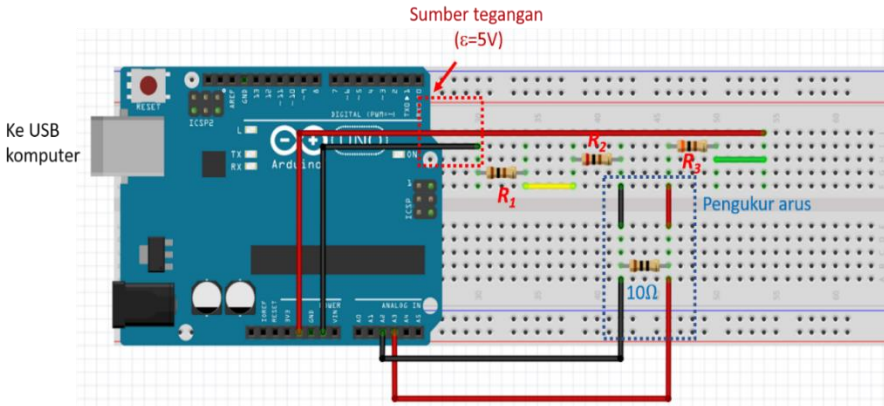
1. Menyiapkan rangkaian pengukur arus listrik dengan menggunakan Arduino (Lihat Gambar 3.8)
2. Menyiapkan rangkaian pengukur tegangan listrik dengan menggunakan Arduino (Lihat Gambar 3.2)
3. Merangkai 3 resistor dan sumber tegangan 5 V pada Arduino secara seri.



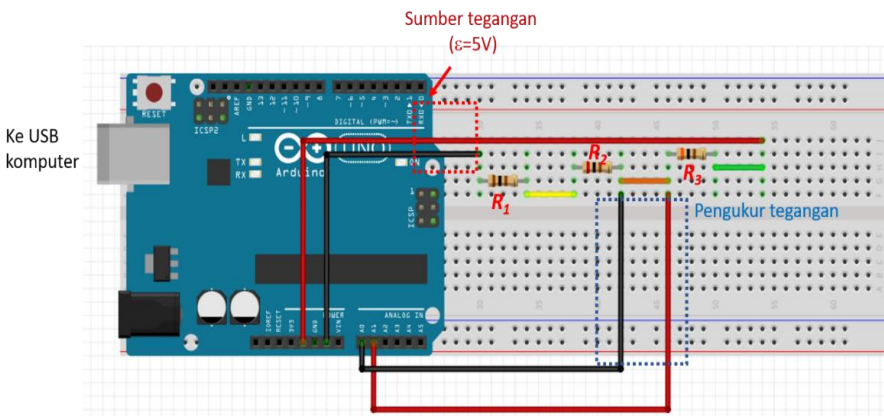
Gambar 4.22. Diagram rangkaian *loop* tunggal pada percobaan Hukum Kirchhoff

4. Mengukur arus yang mengalir pada R_1, R_2 dan R_3 (I_1, I_2 , dan I_3)
5. Mengukur tegangan pada R_1, R_2 dan R_3 (V_1, V_2 , dan V_3) serta ϵ

6. Mengubah sumber tegangan menjadi 3,3 V kemudian mengulangi langkah 3-5.



Gambar 4.23 Contoh rangkaian untuk mengukur I_1 , I_2 , dan I_3

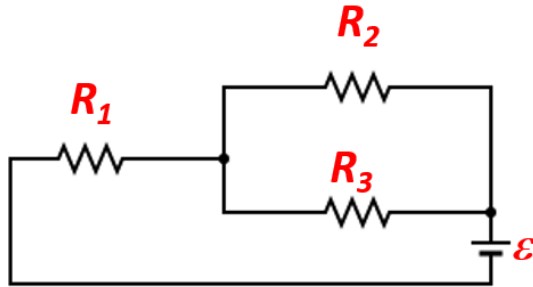


Gambar 4.24 Contoh rangkaian untuk mengukur V_1 , V_2 , dan V_3

Percobaan 2

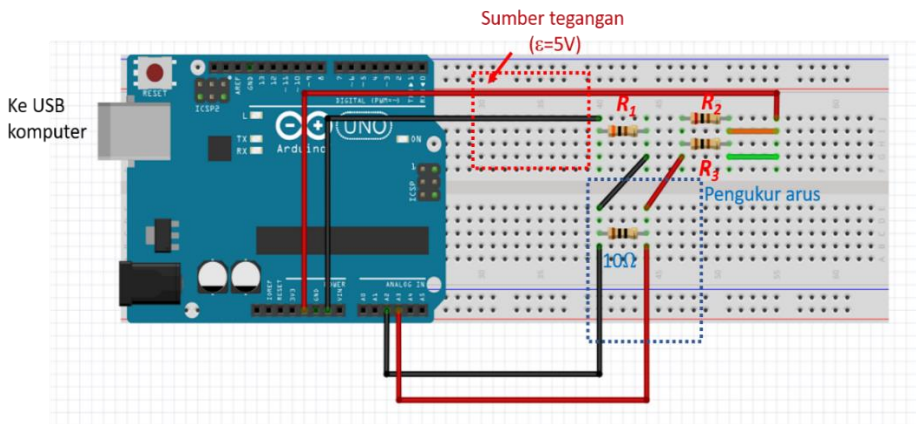
1. Menyiapkan rangkaian pengukur arus listrik dengan menggunakan Arduino (Lihat Gambar 3.8)
2. Menyiapkan rangkaian pengukur tegangan listrik dengan menggunakan Arduino (Lihat Gambar 3.2)

3. Merangkai 3 resistor dan sumber tegangan 5 V menjadi seperti pada Gambar 4.24

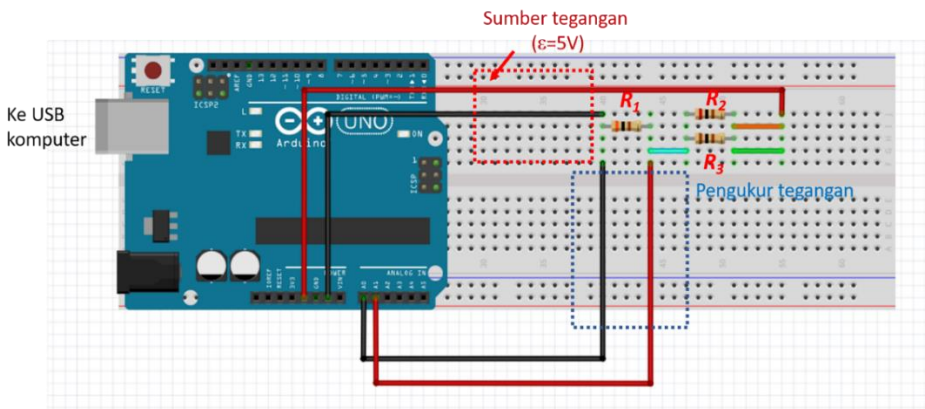


Gambar 4.25 Diagram rangkaian percobaan Hukum II Kirchhoff

4. Mengukur arus yang mengalir pada R_1, R_2 dan R_3 (I_1, I_2, I_3)
5. Mengukur tegangan pada R_1, R_2 dan R_3 (V_1, V_2, V_3) dan ϵ
6. Mengubah sumber tegangan menjadi 3,3 V kemudian mengulangi langkah 3-5.



Gambar 4.26 Contoh rangkaian untuk mengukur $I_1, I_2,$ dan I_3 pada titik percabangan



Gambar 4.27 Contoh rangkaian untuk mengukur V_1 , V_2 , dan V_3 pada rangkaian dengan percabangan



Gambar 4.28 Contoh *script* untuk mengambil data arus (untuk rangkaian pada Gambar 4.23 dan 4.26). Arus dinyatakan dengan satuan mA.



Gambar 4.29 Contoh *script* untuk mengambil data tegangan (untuk rangkaian pada Gambar 4.24 dan 4.27). Tegangan dinyatakan dengan satuan volt.

Tabulasi Data Percobaan

Percobaan 1

ε	R_1	R_2	R_3	I_1	I_2	I_3	V_1	V_2	V_3

Percobaan 2

ε	R_1	R_2	R_3	I_1	I_2	I_3	V_1	V_2	V_3

Contoh Hasil Percobaan

Percobaan 1

ε (volt)	R_1 (Ω)	R_2 (Ω)	R_3 (Ω)	I_1 (A)	I_2 (A)	I_3 (A)	V_1 (volt)	V_2 (volt)	V_3 (volt)
5,00	100	200	300	0,0083	0,0083	0,0083	0,79	1,6	2,61
3,15	100	200	300	0,0049	0,0054	0,0054	0,49	1,01	1,65

$I_1 R_1$ (volt)	$I_2 R_2$ (volt)	$I_3 R_3$ (volt)	$\Sigma I R$ (volt)	ε (volt)
0,83	1,66	2,49	4,98	5,00
0,49	1,08	1,62	3,19	3,15

Percobaan 2

ε (volt)	R_1 (Ω)	R_2 (Ω)	R_3 (Ω)	I_1 (A)	I_2 (A)	I_3 (A)	V_1 (volt)	V_2 (volt)	V_3 (volt)
5	300	200	100	0,0117	0,00391	0,00782	4,8	0,82	0,82
3,3	300	200	100	0,0083	0,00293	0,00538	2,64	0,54	0,54

I_1 (A)	I_2 (A)	I_3 (A)	$I_2 + I_3$ (A)
0,0117	0,00391	0,00782	0,01173
0,0083	0,00293	0,00538	0,00831

Kesimpulan yang Ingin Diperoleh

Percobaan ini mengarahkan peserta didik untuk menemukan konsep:

1. Pada sebuah rangkaian loop tertutup, berlaku hubungan:
$$\Sigma \varepsilon - \Sigma IR = 0$$
2. Pada titik percabangan suatu rangkaian, berlaku hubungan: $\Sigma I_{in} = \Sigma I_{out}$
3. Pada rangkaian seri, besar arus yang mengalir pada setiap titik sama besar
4. Pada rangkaian paralel, besar tegangan pada tiap percabangan sama besar.

F. Percobaan Pengisian dan Pengosongan Kapasitor (Rangkaian RC)

Tujuan

1. Memodelkan perubahan tegangan kapasitor pada saat pengisian
2. Memodelkan perubahan tegangan kapasitor pada saat pengosongan

Bahan

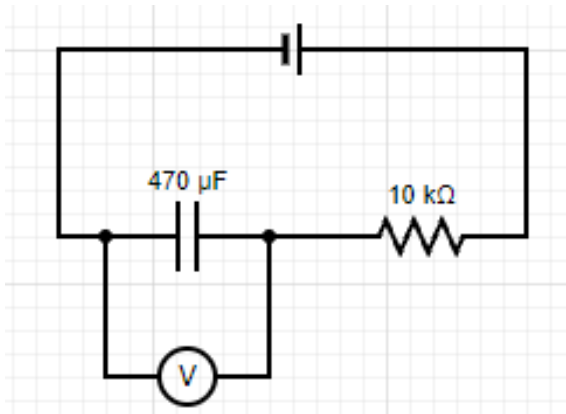
1. Kapasitor 470 μF (1 buah)
2. Resistor 10 $\text{k}\Omega$ (1 buah)
3. *Breadboard* (1 buah)
4. Arduino Uno (1 buah)
5. Kabel Serial USB (1 buah)
6. Kabel *jumper male-male* (4 buah)

Langkah Percobaan

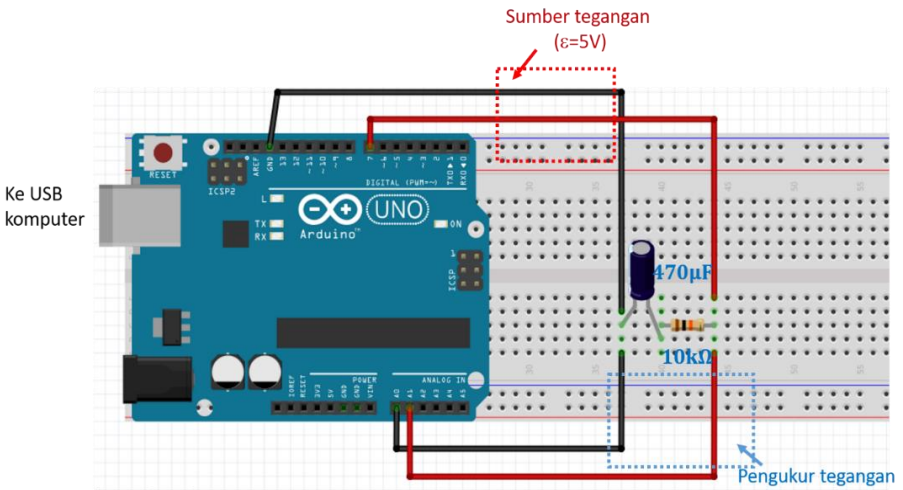
1. Menyiapkan rangkaian pengukur tegangan dengan menggunakan Arduino Uno (Lihat Gambar 3.2)

Pengisian Kapasitor

2. Merangkai kapasitor, resistor, dan sumber tegangan secara seri. Gunakan digital pin 7 sebagai sumber tegangan.



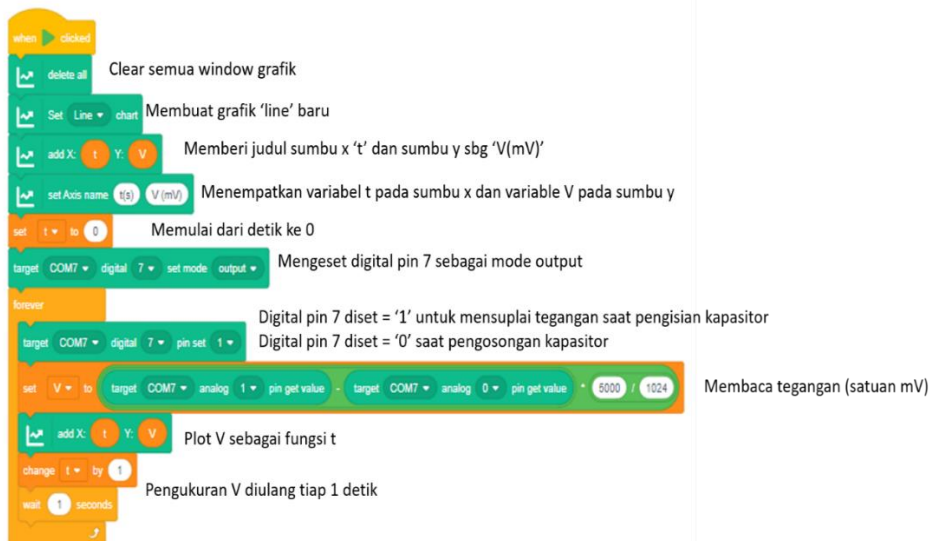
Gambar 4.30 Diagram percobaan rangkaian RC



Gambar 4.31 Contoh realisasi percobaan rangkaian RC

3. Mengukur tegangan pada kapasitor secara otomatis tiap 1 detik dengan menggunakan Arduino.
4. Plot grafik $V-t$ hingga $t=30$ s.

Langkah 3-4 dapat dilakukan otomatis dengan menggunakan *script* pada Gambar 4.32



Gambar 4.32 Contoh *script* untuk mendapatkan plot tegangan terhadap waktu pada percobaan rangkaian RC dengan diagram rangkaian pada Gambar 4.31

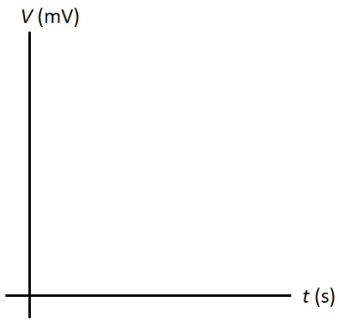
Pengosongan Kapasitor

5. Atur supaya sumber tegangan (digital pin 7) off (0).
6. Mengukur tegangan pada kapasitor secara otomatis tiap 1 detik dengan menggunakan Arduino.
7. Plot grafik V-t hingga $t=35$ s

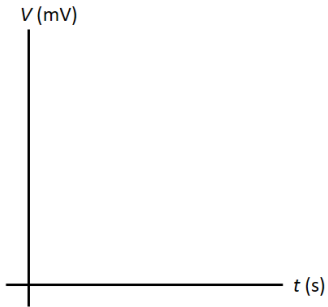
Percobaan pengosongan kapasitor dapat dilakukan setelah percobaan pengisian kapasitor selesai dilakukan. Langkah 5-7 dilakukan otomatis menggunakan sciprt pada Gambar 4.32 dengan mengeset digital pin 7 menjadi 0.

Data Percobaan

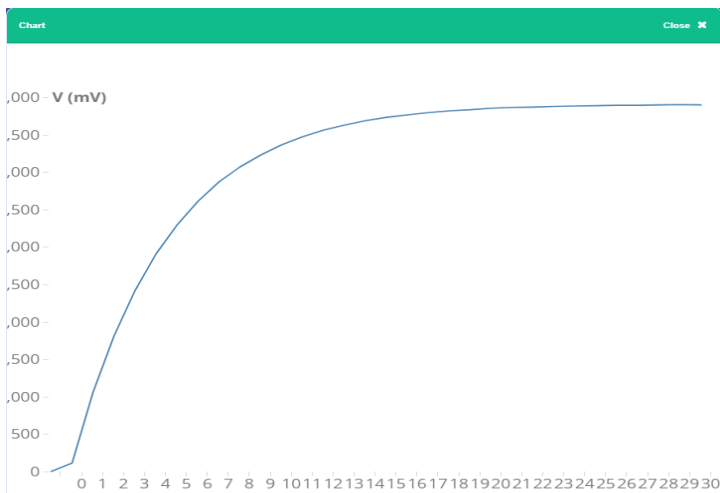
Pengisian Kapasitor



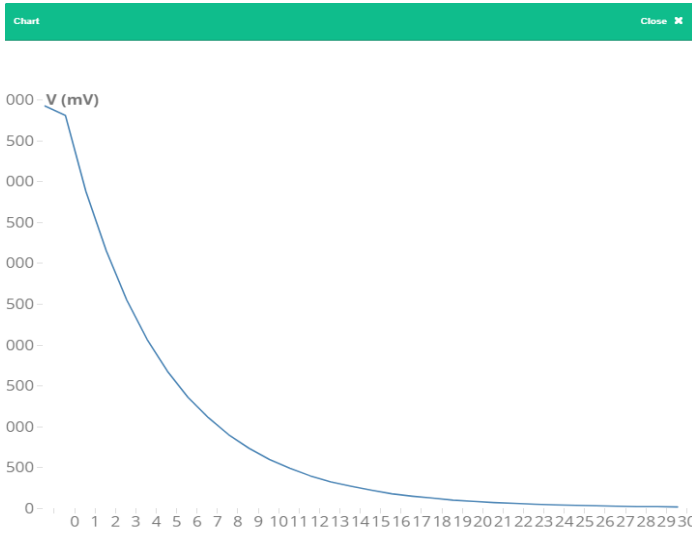
Pengosongan Kapasitor



Contoh Hasil Percobaan



Gambar 4.33 Grafik V - t pada pengisian kapasitor



Gambar 4.34 Grafik V - t pada pengosongan kapasitor

Kesimpulan yang Ingin Diperoleh

Percobaan ini mengarahkan peserta didik untuk menemukan konsep:

1. Pada proses pengisian kapasitor, tegangan kapasitor akan terus naik hingga stabil mendekati nilai tegangan sumber.
2. Pada proses pengosongan kapasitor, tegangan kapasitor akan terus turun secara eksponensial hingga mencapai nol.

DAFTAR PUSTAKA

- Atkin, K. (2018). Investigating the Torricelli law using a pressure sensor with the Arduino and MakerPlot. *Physics Education*, 53, 065001.
- Dijk, B. van. (2019). Voltage Sensor 0-25 V. Retrieved from <https://arduino-tutorials.net/tutorial/reading-voltage-sensor-with-arduino>
- Galeriu, C. (2018). An Arduino Investigation of Newton's Law of Cooling. *The Physics Teacher*, 56, 618. <https://doi.org/10.1119/1.5080580>
- Kadir, A. (2016). *Scratch for Arduino (S4A): Panduan untuk Mempelajari Elektronika dan Pemrograman*. Yogyakarta: Andi Offset.
- Kubínová, Š., & Šlégr, J. (2015). Physics demonstrations with the Arduino board. *Physics Education*, 50(4), 472–474. <https://doi.org/10.1088/0031-9120/50/4/472>
- Pratidhina, E., Rosana, D., Kuswanto, H., & Dwandaru, W. S. B. (2021). Using Arduino and online programming language for physics practical work. *Physics Education*, 56, 055028.
- Yasin, A. I., Prima, E. C., & Sholihin, H. (2018). Learning Electricity using Arduino-Android based Game to Improve STEM Learning Electricity using Arduino-Android based Game to Improve STEM Literacy. *Journal of Science Learning*, 1(3), 77–94. <https://doi.org/10.17509/jsl.v1i3.11789>

GLOSARIUM

Arduino Uno:	Salah satu jenis papan mikrokontroler berbasis ATmega328 yang dikembangkan oleh perusahaan bernama Arduino
<i>Block-Based Programming Language:</i>	Salah satu tipe bahasa pemrograman dimana instruksi kebanyakan direpresentasikan dalam bentuk blok. Sintak program disusun dengan menyusun blok-blok tersebut.
Blok:	Blok adalah bentuk seperti potongan puzzle yang dipakai untuk menyusun kode program pada <i>block-based programming language</i> .
<i>Breadboard:</i>	Sebuah papan yang biasa digunakan untuk merangkai komponen listrik sementara.
Common-Coding:	Salah satu <i>online block-based programming language</i> yang dapat digunakan untuk memprogram Arduino.
<i>Drag and drop:</i>	Memilih dan memindahkan sebuah obyek dengan cara meng-klik obyek dengan tombol mouse sebelah kiri lalu ditahan dan digeser ke tempat baru. Tombol mouse dilepas saat sudah mencapai lokasi baru yang diinginkan.
Hukum Kirchhoff Tentang Arus:	Sebuah pernyataan bahwa jumlah arus yang masuk ke sebuah titik pada rangkaian listrik sama dengan jumlah arus yang keluar dari titik tersebut
Hukum Kirchhoff Tentang Tegangan:	Sebuah pernyataan bahwa jumlah tegangan pada sebuah rangkaian tertutup (<i>loop</i>) sama dengan nol
Hukum Ohm:	Sebuah pernyataan bahwa arus yang mengalir pada konduktor berbanding lurus

	dengan tegangan antara ujung-ujung konduktor.
Isolator Listrik:	Material yang sulit menghantarkan arus listrik (konduktivitas sangat rendah)
Kabel <i>Jumper</i> :	Kabel elektrik yang memiliki pin konektor di setiap ujungnya untuk menghubungkan dua komponen tanpa memerlukan solder.
Kapasitor:	Komponen elektronik yang berfungsi menyimpan muatan
Konduktor Listrik:	Material yang dapat menghantarkan arus listrik dengan baik, memiliki konduktivitas tinggi
<i>Non-ohmic Material</i> :	Bahan yang tidak memenuhi Hukum Ohm
<i>Ohmic Material</i> :	Bahan yang memenuhi Hukum Ohm
Rangkaian Paralel:	Rangkaian listrik yang terdiri dari lebih dari satu jalur untuk melewatkan arus
Rangkaian Seri:	Rangkaian listrik yang terdiri dari hanya satu jalur untuk melewatkan arus
Scratch:	Salah satu <i>block-based programming language</i> yang dikembangkan oleh Massachusetts Institute of Technology.
Semikonduktor:	Material yang pada kondisi tertentu saja dapat menghantarkan arus listrik, konduktivitasnya berada di antara konduktor dan isolator.
<i>Variable Resistor</i> :	Jenis resistor yang dapat diatur nilai hambatannya
<i>Variable Voltage</i> :	Jenis sumber tegangan yang dapat diatur nilai tegangan luarannya

INDEKS

A

Arduino, iii, v, 1, 2, 3, 4, 5, 6, 11, 15, 16, 17, 18, 19, 20, 21, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 47, 48, 54, 55, 62, 63, 68, 69, 74, 75, 76, 79, 81, 83

B

Block-Based Programing Language, 81,82

Blok, 15, 21, 22, 23, 81

Breadboard, 6, 7, 8, 9, 10, 11, 41, 47, 54, 62, 67, 74, 81

C

Common-Coding, iii, 1, 15, 16, 17, 18, 19, 21, 24, 30, 33, 39, 41, 81

D

Drag and drop, 22, 23, 81

H

Hukum Kirchhoff, 41, 67, 68, 81

Hukum Ohm, 41, 42, 43, 47, 61, 81

I

Isolator, 82

K

Kabel *jumper*, 41, 47, 54, 62, 67, 74, 82

Kapasitor, 6, 12, 41,74, 75, 76, 77, 78, 82

Konduktor, 36, 41, 46, 47, 52, 53, 54, 61, 81, 82

N

Non-ohmic, 41, 54, 61, 82

O

Ohmic, 41, 54, 61, 82

P

paralel, 10, 61, 63, 64, 66, 73,82

S

Scratch, 1, 79, 82

Semikonduktor, 61, 82

seri, 4, 10, 33, 41, 47, 54, 61, 62, 63, 64, 65, 66, 67, 68, 73, 74, 82

V

Variable resistor, 82

Variable voltage, 13, 82

CATATAN

Gambar rangkaian pada buku ini dibuat dengan program Fritzing dan Circuit Diagram (<https://www.circuit-diagram.org/>) yang bersifat free.

Tentang Penulis



Elisabeth Pratidhina menyelesaikan pendidikan master di Department of Physics, Korea Advanced Institute of Science and Technology, Republic of Korea. Saat ini sedang menyelesaikan pendidikan doctoral di Universitas Negeri Yogyakarta dan mengajar di Program Studi Pendidikan Fisika, Universitas Katolik Widya Mandala Surabaya, Indonesia.



Heru Kuswanto menyelesaikan pendidikan doctoral pada bidang fisika di Jean Monnet University, France. Saat ini menjadi professor di Jurusan Pendidikan Fisika, Universitas Negeri Yogyakarta. Topik penelitian yang ditekuni adalah optoelektronika dan pembelajaran fisika dengan mobile learning.



Dadan Rosana menyelesaikan pendidikan doctoral pada bidang pendidikan dari Universitas Negeri Yogyakarta. Saat ini menjadi professor di Program Studi Pendidikan Sains, Universitas Negeri Yogyakarta, Indonesia.



 www.ciptapublishing.com
 ciptapublishing@gmail.com
 [ciptapublishing](https://www.instagram.com/cipta_publishing)

ISBN 978-623-5647-08-1

