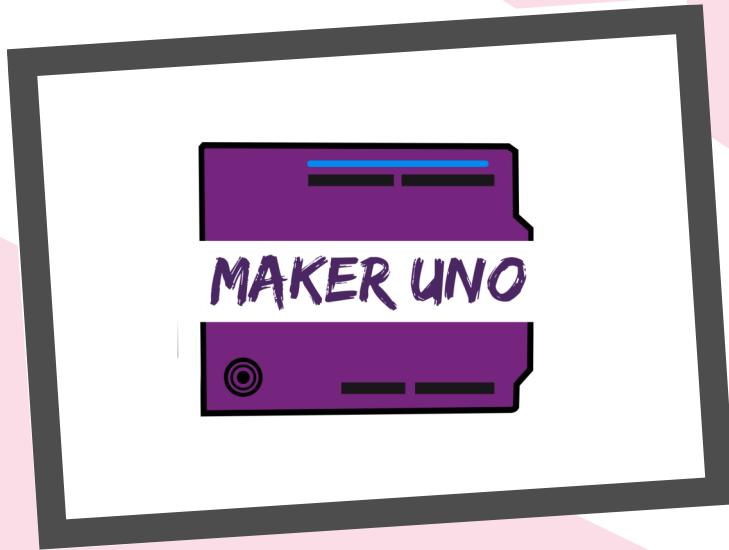


# Panduan Guru Maker UNO/ Arduino



Ditulis khas berdasarkan dokumen  
standard kandungan prestasi subjek  
**Reka Bentuk Teknologi Tingkatan Dua**

# PENDAHULUAN

Modul ini ditulis berdasarkan Dokumen Standard Kurikulum dan Pentaksiran (DSKP) untuk subjek Reka Bentuk Teknologi (RBT) Tingkatan 2 terbitan Kementerian Pendidikan Malaysia.

Modul ini memperkuatkhan dan menyediakan guru dengan pengetahuan dan kemahiran untuk mengajar topik elektronik yang terkandung dalam DSKP RBT Tingkatan 2. Modul ini ditulis khas untuk guru-guru yang tiada pengetahuan khusus dalam pengaturcaraan dan juga penggunaan mikropengawal.

Modul ini direka khas untuk papan Maker UNO. Modul ini juga boleh digunakan untuk semua papan mikropengawal yang setara (*compatible*) dengan Arduino UNO, misalnya CT-UNO. Modul ini juga disertakan dengan set modul murid yang boleh diguna bersama dengan modul guru.

Modul ini dibangun dan ditulis oleh guru, untuk guru; dengan kerjasama pihak Cytron Technologies dan juga Arus Academy. Arus Academy merupakan sebuah perusahaan sosial yang berpusat di Alma, Bukit Mertajam. Arus Academy diasaskan oleh 4 orang guru sekolah menengah dan misinya adalah untuk memberikan pendidikan STEM yang berkualiti dan relevan untuk golongan murid kurang berkemampuan.

Arduino menekankan semangat dan prinsip sumber terbuka (*Open Source*). Modul ini juga diterbit dengan prinsip yang sama. Pengguna digalakkan untuk berkongsi, mengubahsuai dan mengedarkan semula maklumat yang terkandung dalam modul ini. Pengguna cuma perlu memberi kredit yang sewajarnya kepada penulis modul ini.

Modul ini adalah terlindung di bawah lesen *Creative Commons Atribusi-Perkongsian Serupa 4.0 Antarabangsa (CC BY-SA 4.0)*.





Karya ini dilesenkan di bawah Lesen Atribusi-Perkongsian Serupa 4.0 Antarabangsa Creative Commons.

**Anda bebas untuk:**

- **Kongsi** – salin dan edar semula bahan tersebut dalam sebarang medium atau format
- **Adaptasi** – adunan semula, mengubah, dan membina di atas bahan untuk sebarang tujuan, walaupun secara komersial.

**Di bawah syarat-syarat berikut:**

- **Atribusi** – Anda perlu memberikan kredit yang sewajarnya, menyediakan pautan kepada lesen, dan menunjukkan jika ada perubahan yang dibuat. Anda boleh berbuat demikian dengan apa jua cara yang munasabah, tetapi bukan dengan cara yang seolah-olah pemberi lesen yang mengiktiraf anda atau penggunaan anda.
- **Perkongsian Serupa** – Jika anda mengadun semula, mengubah atau mencipta sesuatu berdasarkan material tersebut, anda perlu mengedarkan karya-karya anda mengikut lesen yang sama seperti karya asal.
- **Tiada larangan tambahan** – Anda tidak boleh guna terma-terma perundungan atau langkah-langkah teknologi yang menyekat pengguna lain daripada membuat sesuatu yang dibenarkan oleh lesen.

**Ringkasan Penerbitan**

Versi pertama diterbit pada 24/01/2018

**Untuk sebarang maklumat / pembetulan / cadangan, sila hubungi:**

[enquiry@arusacademy.org.my](mailto:enquiry@arusacademy.org.my)

# ISI KANDUNGAN

CADANGAN PENGAGIHAN MASA PDPC	5
PERSEDIAAN	6
UNIT 1– MIKRO PENGAWAL	7
1.1- APAKAH ITU MIKRO PENGAWAL DAN FUNGSINYA	8
1.2- BAHAGIAN-BAHAGIAN DALAM MIKRO PENGAWAL	11
1.3- GAMBAR RAJAH BLOK	13
1.4- PENGENALAN KEPADA RAJAH SKEMATIK	14
UNIT 2- OUTPUT	17
2.1- PENGENALAN KEPADA PENGATURCARAAN OUTPUT	18
2.2- JENIS-JENIS PERANTI OUTPUT	30
2.3- PENGENALAN KEPADA SAMBUNGAN LITAR OUTPUT	34
2.4- SIMULASI LITAR OUTPUT	42
UNIT 3- INPUT	49
2.1- PENGENALAN KEPADA PENGATURCARAAN INPUT	50
2.1- JENIS-JENIS PERANTI INPUT	54
2.1- PENGENALAN KEPADA SAMBUNGAN LITAR INPUT	59
2.1- SIMULASI LITAR INPUT	63



# ISI KANDUNGAN

UNIT 4- GABUNGAN OUTPUT DAN INPUT	68
2.1- PENGENALAN KPD PENGATURCARAAN STRUKTUR KAWALAN PILIHAN	69
2.1- PENYAMBUNGAN LITAR OUTPUT DAN INPUT	76
2.1- SIMULASI LITAR OUTPUT DAN INPUT	78
2.1- PENGHASILAN PROJEK	83
LAMPIRAN 1-CARA MENGATURCARA DENGAN TELEFON PINTAR	84
LAMPIRAN 2-RUJUKAN TAMBAHAN PENGGUNAAN ARDUINO	85
MODUL MURID	87
UNIT 1- MIKROPENGAWAL	88
UNIT 2- OUTPUT	92
UNIT 3- INPUT	103
UNIT 4- OUTPUT DAN INPUT	111
SKEMA JAWAPAN MODUL MURID	
UNIT 1- MIKROPENGAWAL	118
UNIT 2- OUTPUT	121
UNIT 3- INPUT	131
UNIT 4- OUTPUT DAN INPUT	138



# CADANGAN PENGAGIHAN

## MASA PDPC

UNIT	TOPIK	CADANGAN MASA / WAKTU	CATATAN
UNIT 1	TEORI MIKROPENGAWAL	2 JAM	
UNIT 2	OUTPUT	3 JAM	
UNIT 3	INPUT	2 JAM	SESETENGAH TOPIK PERLU KOMPUTER ATAU/ DAN TELEFON PINTAR
UNIT 4	GABUNGAN OUTPUT / INPUT	3 JAM	

# **PERSEDIAAN**

## **Mikropengawal Arduino**

- Guru boleh belikan set Maker UNO dari kedai atas talian Cytron  
<https://www.cytron.io/p-maker-uno>
- Guru juga boleh belikan set Arduino dari laman sesawang seperti Lelong dan Lazada

## **Perisian Arduino**

- Perisian Arduino boleh dimuat turun dari laman web Arduino rasmi :  
<https://www.arduino.cc/en/Main/Software>

## **Pencetakan Modul**

- Modul guru dan murid adalah direka untuk dicetak dalam bentuk A5 (separuh A4)

# UNIT 1

# MIKROPENGAWAL

## STANDARD PEMBELAJARAN

- 2.4.1 Menyatakan maksud mikropengawal dan mikropemproses
- 2.4.2 Menjelaskan bahagian-bahagian yang terdapat dalam mikropengawal
- 2.4.3 Menghasilkan lakaran reka bentuk litar elektronik.

## STANDARD PENILAIAN

- TAHAP 1 Menyatakan maksud dan bahagian-bahagian yang terdapat dalam mikropengawal
- TAHAP 2 Menerangkan fungsi peranti yang terdapat dalam litar mikropengawal.
- TAHAP 3 Melakar reka bentuk litar elektronik menggunakan mikropengawal.

## SUB-UNIT

UNIT 1.1	Apakah itu mikropengawal dan mikropemproses	SP: 2.4.1
UNIT 1.2	Bahagian dalam mikropengawal	SP: 2.4.2
UNIT 1.3	Pengenalan kepada gambar rajah blok	
UNIT 1.4	Pengenalan kepada rajah skematik	SP: 2.4.3

## CADANGAN PEMBAHAGIAN MASA PDPC

## PERSEDIAAN

- 1) Boleh digunakan berserta dengan modul murid

UNIT 1.1	30 MINIT
UNIT 1.2	
UNIT 1.3	30 MINIT
UNIT 1.4	60 MINIT

## UNIT 1.1

# APAKAH ITU MIKROPENGAWAL DAN MIKROPEMPROSES

### Objektif Pembelajaran

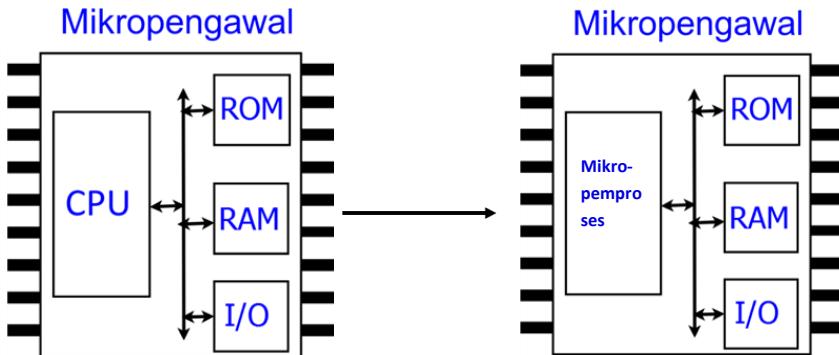
Dalam unit ini, murid akan dapat menyatakan maksud mikropengawal dan mikropemproses

### Kriteria Kejayaan:

Murid dapat menyatakan maksud mikropengawal secara lisan atau bertulis

Mikropengawal adalah peranti kawalan. Peranti kawalan ini menggunakan mikropemproses sebagai unit pemprosesan pusat (CPU), RAM (Random Access Memory), ROM (Read Only Memory) dan juga peranti *Input* dan *Output* (I/O). CPU, RAM, ROM dan peranti I/O berada dalam satu cip yang tunggal.

Mikropemproses adalah unit pemprosesan pusat (CPU) yang wujud sebagai cip tunggal. RAM, ROM dan peranti I/O adalah berasingan dengan CPU.



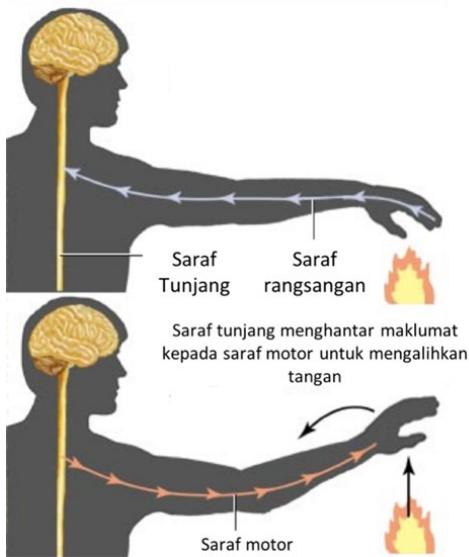
Rajah 1.1.(a): Rajah ini menunjukkan bahawa CPU adalah mikropemproses

Mikropengawal sering digunakan dalam pelbagai peranti di sekeliling kita. Contohnya dalam televisyen, penghawa dingin, dan kereta moden.

# UNIT 1.1

Mikropengawal bertugas untuk menerima *input* (masukan), memproses maklumat yang diterima dan mengeluarkan *output* (keluaran) yang tepat. Mikropengawal boleh dianggap sebagai sebuah komputer yang kecil atau otak.

Perbandingan boleh dibuat dengan rajah 1.1(b) di atas. Sistem mikropengawal bertindak seperti sistem saraf manusia. Saraf rangsangan menerima input dari tangan dan menghantar maklumat tersebut ke saraf tunjang untuk diproses dalam otak.



Rajah 1.1.(b): Rajah sistem saraf manusia.

Sebuah mikropengawal boleh diberi arahan melalui pengaturcaraan untuk melakukan tugas secara automatik.

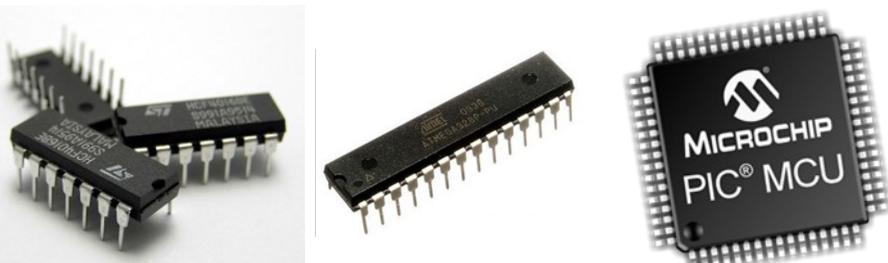
Selepas diberi arahan, sebuah mikropengawal akan menyimpan arahan tersebut tanpa memerlukan penyambungan dengan sebuah komputer.

Scara rumusan, mikropengawal berfungsi seperti rajah 1.1(c) iaitu mendapatkan maklumat (*input*) dan bertindakbalas (*output*) berdasarkan arahan yang telah ditetapkan.



Rajah 1.1.(c): Gambar rajah blok bagi mikropengawal .

# UNIT 1.1



Rajah 1.1.(d): Jenis mikropengawal yang wujud

Antara papan pembangunan mikropengawal (*Microcontroller Development Board*) yang sering digunakan adalah seperti berikut:



Rajah 1.1.(e): Intel Edison



Rajah 1.1.(f): Arduino UNO



Rajah 1.1.(g): Maker UNO



Rajah 1.1.(h): Beaglebone Black

## UNIT 1.2

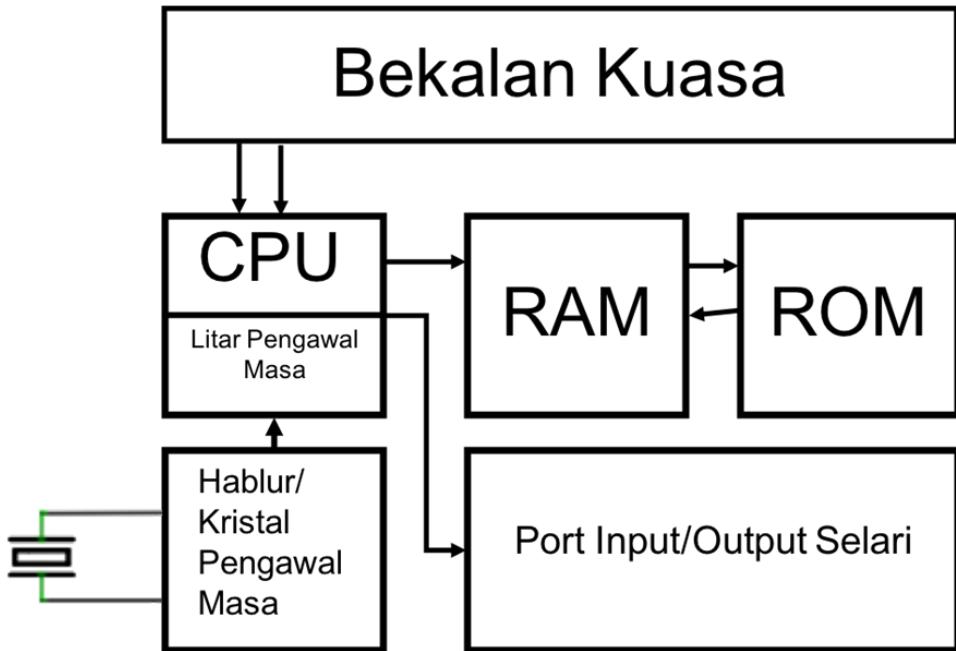
# BAHAGIAN-BAHAGIAN DALAM MIKROPENGAWAL

### Objektif Pembelajaran

Dalam unit ini, murid akan dapat menjelaskan bahagian-bahagian dalam mikropengawal

### Kriteria Kejayaan:

Murid dapat menyatakan bahagian-bahagian dalam mikropengawal secara lisan atau bertulis



Rajah 1.2.(a): Rajah menunjukkan bahagian-bahagian dalam mikropengawal

# UNIT 1.2

Bahagian	Fungsi
CPU	Menerima maklumat dan arahan(program) untuk memproses <i>input</i> dan <i>output</i> .
RAM & ROM	Ruang memori untuk menyimpan maklumat dan arahan (program).
Port Input/Output Selari	Menyambung kepada peranti <i>input</i> dan <i>output</i> seperti LED, motor & penderia ( <i>sensor</i> ). Terdapat 2 jenis isyarat <i>Input</i> dan <i>Output</i> iaitu isyarat analog dan isyarat digital.
Litar Pengawal Masa	Memberi keupayaan mikropengawal untuk mengawal sistem berdasarkan masa.
Habbur/Kristal Pengawal Masa	Digunakan untuk menghasilkan frekuensi yang digunakan dalam litar pengawal masa.
Bekalan Kuasa	Memberi bekalan tenaga elektrik kepada mikropengawal.

## UNIT 1.3

# PENGENALAN KEPADA GAMBAR RAJAH BLOK

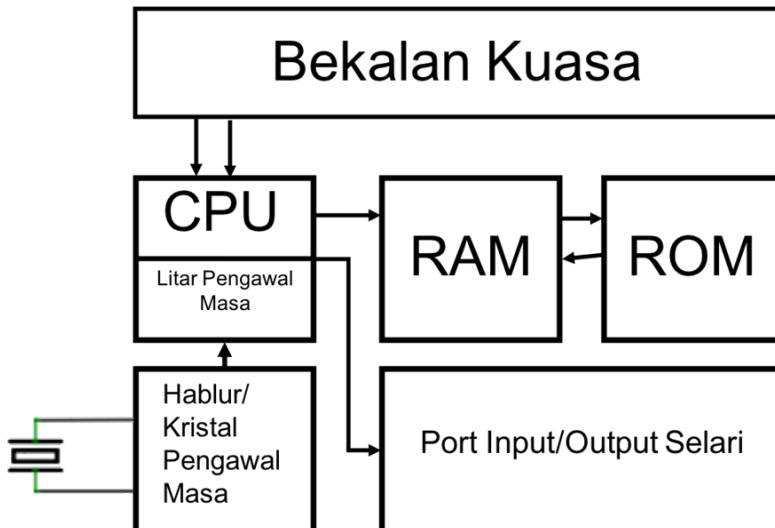
### Objektif Pembelajaran

Dalam unit ini, murid akan dapat mengenali cara melukis gambar rajah blok

### Kriteria Kejayaan:

Murid dapat melukis satu contoh gambar rajah blok yang mengandungi bekalan kuasa, mikropengawal, input dan output

Rajah berikut menunjukkan satu contoh gambar rajah blok.



Rajah 1.3.(a): Gambar rajah blok bagi mikropengawal

## UNIT 1.4

# PENGENALAN KEPADA GAMBAR RAJAH SKEMATIK

### Objektif Pembelajaran

Dalam unit ini, murid akan dapat mengenali cara melukis gambar rajah skematik

### Kriteria Kejayaan:

Murid dapat melukis rajah skematic dengan tepat berlandaskan panduan yang diberi

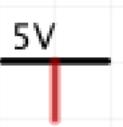
---

Garis panduan untuk menghasilkan lakaran reka bentuk litar elektronik.

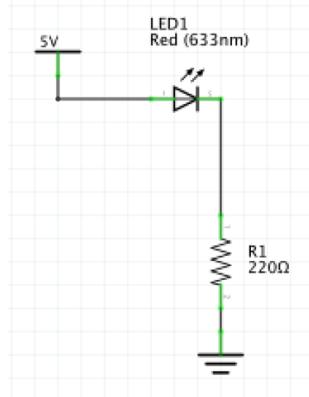
1. Pastikan garisan dilukis dengan lurus
2. Pastikan garisan yang digunakan bukanlah anak panah.
3. Pastikan garisan dilakarkan menegak atau melintang sahaja.
4. Kurangkan pintasan garisan bagi mengelakkan kekeliruan
5. Gunakan simbol yang tetap bagi menggambarkan setiap komponen.
6. Pastikan setiap komponen dalam rajah skematic dilabelkan

# UNIT 1.4

Elemen - elemen penting dalam gambar rajah skematik:

Bahagian	Fungsi
 Ground / Bumi (GND)	Merupakan terminal negatif dalam sesebuah litar. Secara umumnya, litar elektronik akan berakhir dengan GND.
 Bekalan Kuasa 5V	Permulaan litar elektronik. Sumber bekalan kuasa.

Berikut adalah contoh gambar rajah skematik litar elektronik.



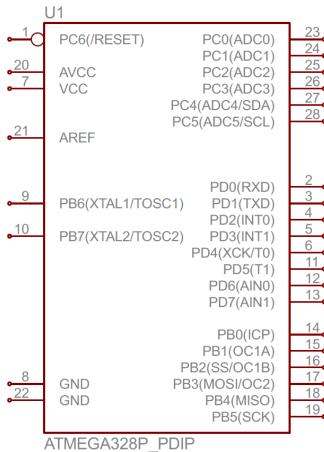
Perhati bahawa litar ini memenuhi syarat

- Garisan dilukis lurus
- Garisan yang digunakan bukanlah anak panah.
- Garisan dilakarkan menegak atau melintang sahaja.
- Tiada pintasan garisan.
- Gunakan simbol yang tetap bagi menggambarkan setiap komponen.
- Pastikan setiap komponen dalam gambar rajah skematik dilabelkan.

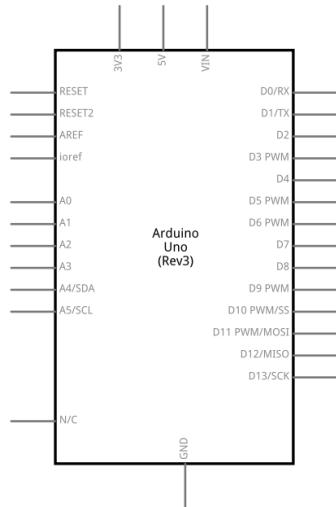
Rajah 1.4.(a): Gambar rajah skematik

# UNIT 1.4

Berikut adalah simbol untuk mikropengawal:

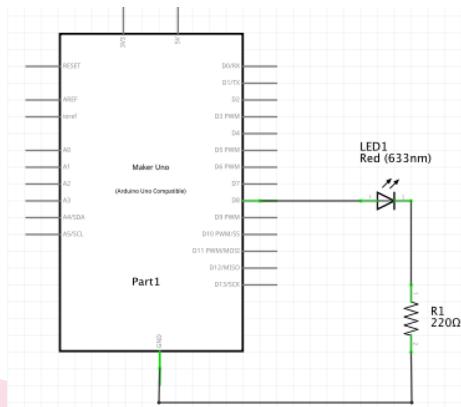


Rajah 1.4.(b): Gambar rajah skematic untuk cip ATMEGA328P (mikropengawal dalam papan Arduino Uno)



Rajah 1.4.(c): Gambar rajah skematic papan mikropengawal Arduino Uno

Berikut adalah contoh gambar rajah skematic yang menggunakan mikropengawal.



Rajah 1.4.(d): Rajah blok bagi sistem LED yang bersambung ke pin D8 dan berakhir di GND

# UNIT 2

# OUTPUT

## STANDARD PEMBELAJARAN

- 2.4.4 Membina litar simulasi yang berfungsi dengan perisian khas
- 2.4.5 Membuat penyambungan litar *input* dan *output* kepada mikropengawal
- 2.4.6 Menulis pengaturcaraan mudah berdasarkan penyambungan litar *input* dan *output*

## STANDARD PENILAIAN

TAHAP 4 Menguji kefungsian litar yang menggunakan mikropengawal (*microcontroller*).

## SUB-UNIT

UNIT 2.1	Pengenalan kepada pengaturcaraan litar <i>output</i>	SP: 2.4.6
UNIT 2.2	Jenis peranti <i>output</i>	
UNIT 2.3	Pengenalan kepada penyambungan litar <i>output</i>	SP: 2.4.5
UNIT 2.4	Simulasi litar <i>output</i>	SP: 2.4.5

## CADANGAN PEMBAHAGIAN MASA PDPC

UNIT 2.1	60 MINIT
UNIT 2.2	
UNIT 2.3	60 MINIT
UNIT 2.4	60 MINIT

## PERSEDIAAN

- 1) Pastikan ada set Arduino / Maker UNO yang mencukupi untuk murid
- 2) Unit 2.1 memerlukan komputer atau telefon pintar untuk pengaturcaraan. (Rujuk Lampiran 1)
- 3) Unit 2.4 memerlukan komputer dan Internet yang berfungsi
- 4) Boleh digunakan bersama dengan modul murid

## UNIT 2.1

# PENGENALAN KEPADA PENGATURCARAAN OUTPUT

### Objektif Pembelajaran

Dalam unit ini, murid akan dapat menulis atur cara mudah bagi mengawal litar output.

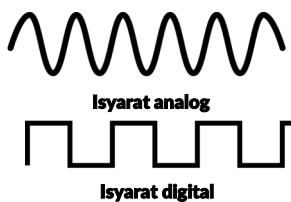
### Kriteria Kejayaan:

Murid dapat mengawal LED bina dalam untuk Arduino / Maker UNO

Mikropengawal boleh berfungsi untuk mengawal litar keluaran (*output*) dengan menghantar isyarat kepada pin keluaran di mikropengawal. Dengan ini, komponen-komponen elektronik yang bersambung pada pin mikropengawal dapat dikawal melalui pengaturcaraan.

Secara umum, terdapat 2 jenis isyarat yang boleh dikeluarkan daripada mikropengawal, iaitu isyarat digital dan isyarat analog. Sebelum bermula menulis sebuah pengaturcaraan, kita perlu memahami perbezaan antara isyarat digital dan isyarat analog.

Secara ringkasnya, isyarat digital mempunyai 2 keadaan (*state*), iaitu *ON* (litar mempunyai arus elektrik) atau *OFF* (tiada arus elektrik dalam litar). Dengan menghantar isyarat digital *ON* kepada pin 5, kita boleh mengaktifkan komponen yang bersambung pada pin 5. Manakala jika kita menghantar isyarat digital *OFF* kepada pin 5, maka komponen yang bersambung pada pin 5 akan dinyahaktifkan / dipadamkan.

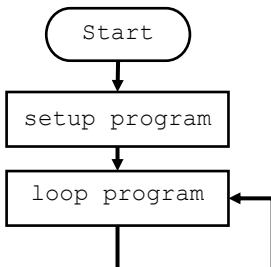


Rajah 2.1.(a): Perbandingan antara isyarat digital dan analog

Bagi isyarat analog pula, ia mempunyai lebih daripada 2 nilai, dan untuk mikropengawal Arduino, ia merangkumi 256 nilai bermula daripada 0 hingga ke 255. Nilai 0 bermaksud tiada arus elektrik (nilai voltan 0) dan nilai 255 adalah nilai maksimum. Nilai isyarat analog (0 hingga 255) adalah berkadar terus dengan nilai voltan yang dihasilkan pada pin keluaran (contohnya nilai 127 akan menghasilkan 50% daripada voltan maksimum). Untuk lebih mudah memahami, jika kita menyambungkan LED pada pin keluaran analog, nilai 0 akan memadamkan LED, nilai 127 menjadikan LED menyala dengan 50% keterangan, dan nilai 255 akan menyalakan LED dengan 100% keterangan.

# UNIT 2.1

Mikropengawal Arduino menjalankan dua fungsi utama, iaitu fungsi *setup* dan fungsi *loop*. Fungsi *setup* akan dilaksanakan sekali sahaja setelah Arduino dihidupkan atau *reset*, manakala fungsi *loop* akan dijalankan berulang kali sehingga Arduino dipadamkan.

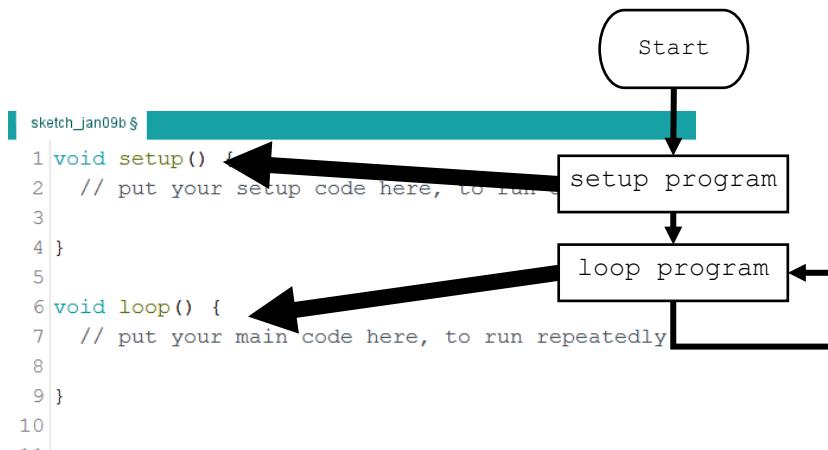


Rajah 2.1.(b): Carta alir untuk Arduino

Aturcara untuk Arduino boleh ditulis pada Arduino IDE (*Integrated Development Environment*) iaitu perisian pembangunan arut cara bagi Arduino. Perisian tersebut boleh dimuat turun dari laman web rasmi Arduino [www.arduino.cc](http://www.arduino.cc).

Setelah membuka perisian Arduino IDE, anda digalakkan untuk memaparkan nombor baris bagi memudahkan pengaturcaraan. Klik pada *File - Preferences*, tandakan pada *Display Line Number*. Klik OK.

Anda akan dapati antaramuka perisian Arduino adalah seperti di rajah 2.1(c). Perhatikan dua fungsi yang telah dibina secara automatik pada *sketch Arduino - setup* dan *loop*.



Rajah 2.1.(c): Perbandingan antara carta alir Arduino dengan arut cara Arduino IDE

# UNIT 2.1

Langkah-langkah menulis atur cara Arduino:

## Langkah 1–Tetapkan pinMode di void setup()

Kenalpasti nombor pin yang akan digunakan berserta mod pin tersebut, sama ada mod masukan (*input*) ataupun mod keluaran (*output*)



## Langkah 2–Merancang aturcara anda

Fikirkan process apa yang perlu dicapai dengan pengaturcaraan anda. Buat lakaran carta alir jika perlu.



## Langkah 3–Tuliskan atur cara di fungsi void loop()

Tuliskan atur cara di fungsi loop. Atur cara anda boleh melibatkan proses penghantaran isyarat keluaran, penerimaan, pembacaan isyarat masukan, dan sebagainya.

Peraturan utama pengaturcaraan Arduino:

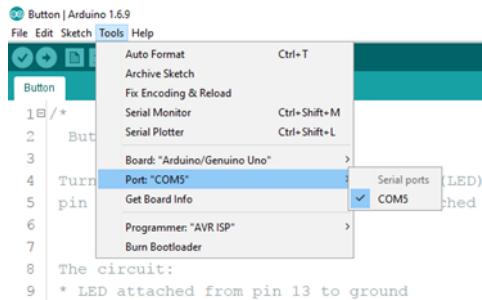
- Pastikan bahawa setiap baris atur cara diakhiri dengan tanda ; (semicolon).
- Pastikan ejaan kata kekunci (keyword) adalah betul (warna **jingga/biru**).
- Pastikan setiap tanda () dan {} berpasangan.

# UNIT 2.1

Langkah-langkah menulis atur cara Arduino:

## Langkah 4-Sambungkan papan Arduino ke komputer menggunakan kabel USB.

Sambungkan board Arduino ke komputer menggunakan kabel USB, dan pastikan Port yang betul dipilih.



## Langkah 5-Tekan ikon Upload (anak panah) dan simpankan fail dengan nama yang sesuai.

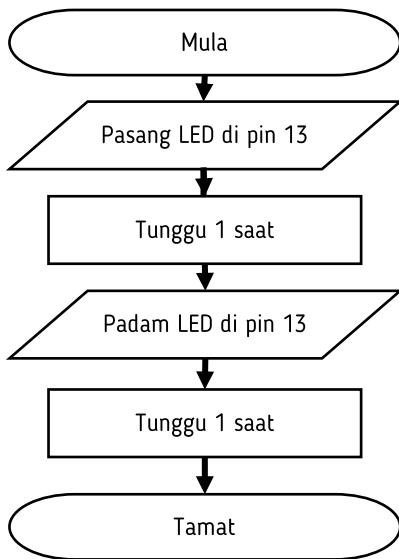


# UNIT 2.1

Setelah kod telah dimuatnaik ke Arduino, kod arahan itu akan kekal dalam memori Arduino sehingga program dipadam / ditulis semula.

Sebelum kita menulis sesuatu atur cara, kita perlu merancangkan atur cara dengan menggunakan carta alir.

Berikut adalah atur cara yang akan dibangunkan.



Ini merupakan satu contoh struktur kawalan jujukan – iaitu atur cara yang tidak mempunyai cabang. Perhatikan bentuk yang berlainan digunakan untuk mewakili langkah atur cara yang berlainan.

Bentuk mewakili permulaan dan penamatkan sesuatu atur cara.

Bentuk mewakili *input* atau *output* pada sesuatu atur cara.

Bentuk mewakili langkah-langkah yang perlu diproses dalam atur cara.

# UNIT 2.1

Berikut adalah atur cara untuk carta alir di atas

```

1 void setup() {
2   pinMode(13,OUTPUT);
3 }
4
5 void loop() {
6   digitalWrite(13,HIGH);
7   delay(1000);
8   digitalWrite(13,LOW);
9   delay(1000);
10 }
```

Fungsi untuk setiap satu baris adalah seperti berikut:

Baris	Pernyataan atur cara	Fungsi
1	void setup()	<p>Menyatakan permulaan untuk fungsi setup. Program setup biasanya digunakan untuk tetapan asas seperti menetapkan mod untuk pin yang digunakan.</p> <p>Perhatikan bahawa perkataan void dan setup adalah dalam warna yang berlainan, ini kerana mereka merupakan kata kekunci khas (keyword) dalam pengaturcaraan Arduino .</p> <p>Simbol { menandakan permulaan untuk program setup.</p>

# UNIT 2.1

Baris	Pernyataan atau cara	Fungsi
2	<code>pinMode(13,OUTPUT);</code>	<p>Menetapkan mod pin tertentu sama ada kepada mod masukan atau mod keluaran</p> <p>Kod ini menetapkan pin 13 sebagai mod keluaran.</p>
3	<code>}</code>	<p>Simbol } menandakan tamat fungsi setup.</p> <p>Bermula dengan simbol { pada baris 1, fungsi <i>setup</i> hanya merangkumi kod di antara baris 1 dan 3 sahaja.</p>
5	<code>void loop() {</code>	<p>Menandakan permulaan untuk fungsi loop.</p> <p>Fungsi <i>loop</i> adalah fungsi yang dijalankan berulang kali sehingga Arduino dipadamkan.</p> <p>Sila perhatikan warna pada perkataan <i>void</i> dan <i>loop</i>, kedua-duanya berlainan – sama seperti <i>setup</i>, perkataan <i>loop</i> juga merupakan kata kekunci dalam Arduino.</p> <p>Simbol { menandakan permulaan untuk fungsi <i>loop</i></p>

# UNIT 2.1

Baris	Pernyataan atur cara	Fungsi
6	<code>digitalWrite(13, HIGH);</code>	<p>DigitalWrite menghantar isyarat digital kepada pin yang dnyatakan.</p> <p>Baris ini akan menghantarkan nilai HIGH kepada pin 13, iaitu akan menghidupkan komponen-komponen yang bersambung ke pin 13.</p> <p>Arduino adalah sensitif kepada huruf besar dan huruf kecil – W untuk digitalWrite dan HIGH perlu ditulis dalam huruf besar baru ia akan dikenalpasti sebagai kata kunci dan akan dipaparkan dalam warna lain.</p>
7	<code>delay(1000)</code>	<p>Delay akan menunggu untuk jangka masa yang ditetapkan. Nilai yang ditulis dalam kurungan delay tersebut adalah dalam unit millisaat. Sebagai contoh, 1000 millisaat bersamaan 1 saat.</p> <p>Jadi delay(1000) bermaksud sistem akan menunggu selama 1 saat.</p>
8	<code>digitalWrite(13, LOW)</code>	<p>Baris ini akan menghantar nilai LOW pada pin 13. Ini akan memadamkan / menyahaktifkan komponen-komponen yang bersambung ke pin 13.</p>

# UNIT 2.1

Baris	Pernyataan atur cara	Fungsi
6	<code>digitalWrite(13, HIGH);</code>	<p>DigitalWrite menghantar isyarat digital kepada pin yang dnyatakan.</p> <p>Baris ini akan menghantarkan nilai HIGH kepada pin 13, iaitu akan menghidupkan komponen-komponen yang bersambung ke pin 13.</p> <p>Arduino adalah sensitif kepada huruf besar dan huruf kecil – W untuk digitalWrite dan HIGH perlu ditulis dalam huruf besar baru ia akan dikenalpasti sebagai kata kunci dan akan dipaparkan dalam warna lain.</p>
7	<code>delay(1000)</code>	<p>Delay akan menunggu untuk jangka masa yang ditetapkan. Nilai yang ditulis dalam kurungan delay tersebut adalah dalam unit millisaat. Sebagai contoh, 1000 millisaat bersamaan 1 saat.</p> <p>Jadi delay(1000) bermaksud sistem akan menunggu selama 1 saat.</p>
8	<code>digitalWrite(13, LOW)</code>	<p>Baris ini akan menghantar nilai LOW pada pin 13. Ini akan memadamkan / menyahaktifkan komponen-komponen yang bersambung ke pin 13.</p>

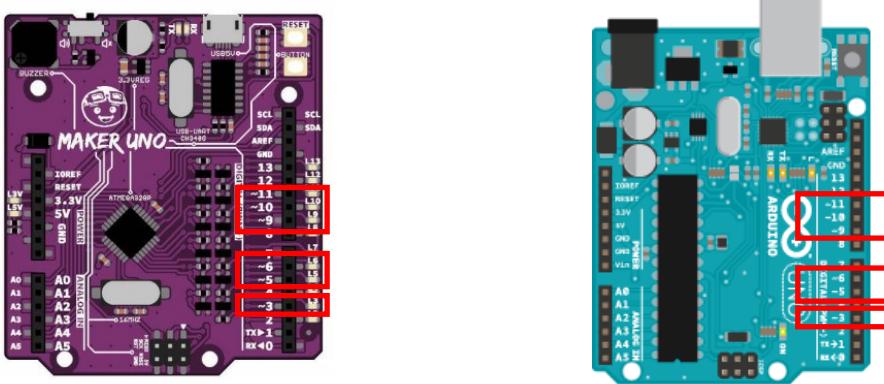
# UNIT 2.1

Baris	Pernyataan atur cara	Fungsi
9	<code>delay(1000)</code>	Kita perlu letak satu lagi delay selepas memadamkan pin 13. Tanpa baris ini, selepas baris 8 dijalankan, aliran aturcara akan kembali ke baris 6.  Walaupun pin 13 akan dipadamkan, tetapi ia akan padam dengan terlalu cepat dan ia tidak dapat dikesan oleh mata kasar kita, jadi ia seolah-olah seperti tidak padam.
10	<code>}</code>	Simbol } menandakan tamat fungsi <i>loop</i> . Bermula dengan simbol { pada baris 5, fungsi <i>loop</i> hanya merangkumi kod di antara baris 5 dan 10 sahaja.

Dengan ini, kita boleh mengawal pin 13 untuk dihidupkan dan dipadamkan setiap saat. Anda akan perhatikan LED atas papan Arduino akan berkelip. Ini kerana pin 13 juga merupakan pin untuk LED yang ada pada Arduino.

# UNIT 2.1

Anda juga boleh menggunakan `analogWrite` untuk menghantar isyarat kepada pin-pin pada Arduino. Bukan semua pin Arduino mampu mengeluarkan isyarat analog, hanya pin yang mempunyai simbol ~ pada papan Arduino sahaja yang boleh mengeluarkan isyarat analog, iaitu pin 9, 10, 11, 3, 5, 6 dan 7.



Rajah 2.1.(d): Lokasi pin analog output yang juga dikenali sebagai pin PWM

Arahan untuk `analogWrite` adalah berikut:

**`analogWrite(pin, nilai)`** di mana nilai adalah antara 0 ke 255

Contohnya,

- `analogWrite(9, 255)` akan menghantar nilai 255 iaitu voltan maksima kepada pin 9. Sebarang komponen yang disambung ke pin 9 akan menerima voltan maksima. Jika LED dipasang di pin 9, ia akan menyala dengan 100% keterangan.
- `analogWrite(9, 127)` akan menghantar nilai 127 (iaitu voltan separuh maksima ke pin 9). Sebarang komponen yang menyambung ke pin 9 akan terima voltan separuh daripada maksima dan LED yang dipasang di pin tersebut akan menyala dengan 50% keterangan
- `analogWrite(9, 0)` akan menghantar nilai 0 (iaitu voltan 0) ke pin 9. Sebarang komponen yang disambung ke pin 9 tidak akan menerima arus kerana voltan adalah 0. LED yang disambung ke pin 9 akan dipadamkan.,

# UNIT 2.1

**Bahagian berikut adalah khusus untuk Maker UNO sahaja. Untuk Arduino yang lain, LED perlu disambungkan pada litar.**

Maker UNO mempunyai LED pada setiap pin digital. Anda boleh menulis arut cara untuk mengawal LED pada setiap pin.

Contohnya:

```
digitalWrite(9, LOW);  
digitalWrite(8, HIGH);  
delay(1000);  
digitalWrite(9, HIGH);  
digitalWrite(8, LOW);  
delay(1000);
```

Arut cara tersebut akan membolehkan LED berkelipl pada pin 9 dan 8 secara bergilir-gilir.

## UNIT 2.2

# JENIS PERANTI OUTPUT

### Objektif Pembelajaran

Dalam unit ini, murid akan dapat mengenali pelbagai jenis peranti output

### Kriteria Kejayaan:

Murid dapat menyenaraikan nama dan fungsi pelbagai jenis peranti output

Peranti *output* akan melaksanakan isyarat yang diberi oleh pin pada Arduino. Arduino boleh membekalkan voltan 5v untuk menghidupkan peranti-peranti yang bersambung pada Arduino. Peranti *output* yang biasa digunakan adalah diod pemancar cahaya dan pembaz. Peranti *output* ini boleh dihidupkan tanpa bekalan kuasa tambahan. Terdapat juga peranti-peranti *output* lain yang perlukan bekalan kuasa tambahan seperti motor arus terus 12v, lampu LED 12v dan sebagainya.

## Perintang

- Perintang digunakan untuk mengehadkan pengaliran arus. Semakin tinggi nilai perintang, semakin banyak rintangan untuk arus melaluinya.
- Perintang adalah penting untuk mengelakkan peranti-peranti daripada rosak atau sebagai pembahagi voltan .
- Nilai perintang dapat dibaca dengan memperhatikan warna jalur pada perintang.
- Simbol perintang pada rajah skematik adalah seperti berikut: 

Perintang 

Warna	Jalur Ke 1 (Nilai Pertama)	Jalur Ke 2 (Nilai Kedua)	Jalur ke 3 (Pendarab)	Jalur Ke 4 (Toleransi)
Hitam	0	0	1 $\Omega$	$\pm 1\%$
Perang	1	1	10 $\Omega$	$\pm 2\%$
Merah	2	2	100 $\Omega$	
Jingga	3	3	1000 $\Omega$	
Kuning	4	4	10000 $\Omega$	
Hijau	5	5	100000 $\Omega$	$\pm 0.5\%$
Biru	6	6	1000000 $\Omega$	$\pm 0.25\%$
Ungu	7	7	10000000 $\Omega$	$\pm 0.10\%$
Kelabu	8	8	100000000 $\Omega$	$\pm 0.05\%$
Putih	9	9	1000000000 $\Omega$	
Emas			0.1 $\Omega$	$\pm 5\%$
Perak			0.01 $\Omega$	$\pm 10\%$

Rajah 2.2.(a): Cara membaca nilai perintang

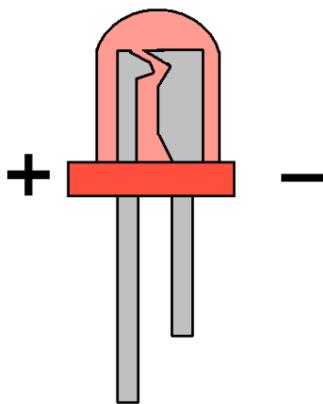


Rajah 2.2.(b): Gambar perintang 10,000 Ohm

# UNIT 2.2

## Diod Pemancar Cahaya (LED)

- Diod pemancar cahaya atau LED (Light Emitting Diode) merupakan diod yang mengeluarkan cahaya apabila arus melaluinya.
- LED boleh digunakan sebagai lampu hiasan dan perlu digunakan bersama dengan perintang untuk mengelakkannya daripada terbakar.
- LED mempunyai kekutuhan pada penyambungannya, iaitu bahagian positif dan negatif.
- Untuk Maker UNO, terdapat LED yang terbina (*built-in*) pada setiap pin digital
- Simbol LED adalah seperti berikut:



Rajah 2.2.(d): Gambar LED

Rajah 2.2.(c): Cara mengenalpasti kaki positif dan negatif untuk LED

# UNIT 2.2

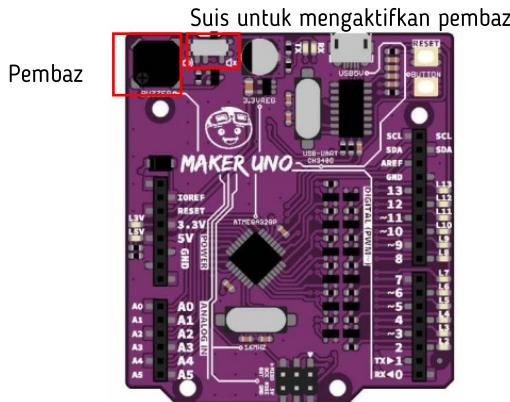
## Pembaz

- Pembaz berfungsi dengan menukarkan isyarat elektrik kepada bunyi.
- Seperti LED, pembaz juga mempunyai kekutuhan pada penyambungannya.
- Pembaz boleh digunakan sebagai penggera keselamatan ataupun memainkan melodi muzik.
- Simbol untuk pembaz adalah seperti berikut:



Rajah 2.2.(e): Gambar pelbagai jenis pembaz

Pada Maker UNO, terdapat satu pembaz yang terbina dalam (*built in*). Pembaz pada Maker UNO bersambung pada pin 8 melalui suis slaid (*sliding switch*). Pembaz itu perlu diaktifkan menggunakan suis slaid sebelum ianya boleh digunakan.



Rajah 2.2.(f): Lokasi pembaz dan suis pembaz pada Maker UNO

# UNIT 2.2

## Motor Arus Terus

- Terdapat pelbagai jenis motor yang memerlukan voltan yang berlainan, contohnya 3v, 6v, 9v, 12v dan selebihnya.
- Aci motor arus terus akan berputar apabila terdapat arus elektrik melaluiinya.
- Secara umumnya, motor perlu bersambung dengan pemandu, ataupun dengan transistor dan diod kerana ia perlukan arus yang tinggi.
- Pemandu (*driver*) yang dipasang pada motor membolehkan kita mengawal arah pusingan motor sama ada lawan jam ataupun ikut arah jam.
- Untuk Maker UNO, satu motor arus terus 3v disediakan. Maker UNO tidak boleh menghidupkan motor yang memerlukan lebih daripada 5v.
- Motor perlu digunakan bersama dengan transistor kerana pin Arduino tidak mampu mengeluarkan arus yang mencukupi untuk menghidupkan motor.
- Simbol untuk motor arus terus adalah seperti berikut:



Rajah 2.2.(g): Gambar pelbagai jenis motor

## UNIT 2.3

# PENGENALAN KEPADA PENYAMBUNGAN LITAR OUTPUT

### Objektif Pembelajaran

Dalam unit ini, murid akan dapat membaca rajah skematik dan menyambungkan litar berdasarkan rajah skematik

### Kriteria Kejayaan:

Murid dapat menghasilkan sekurang-kurangnya 1 litar berdasarkan rajah skematik yang diberi

Penyambungan litar output akan menggunakan wayar pelompat dan juga papan reka (breadboard). Papan reka merupakan papan elektronik yang membolehkan kita menguji litar yang sedang dibina dengan menggunakan wayar pelompat yang tidak memerlukan pateri.

Sebelum kita membina, kita perlu fahami fungsi peranti-peranti berikut:

### Wayar Pelompat

- Wayar pelompat digunakan untuk menyambungkan pelbagai peranti pada mikropengawal.
- Secara tidak rasminya, wayar pelompat merah digunakan untuk wayar yang menyambung ke bekalan kuasa dan wayar pelompat hitam digunakan untuk wayar yang menyambung ke bumi.

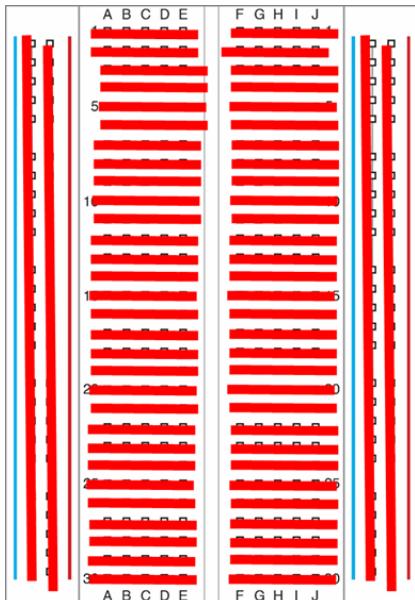


Rajah 2.3.(a): Gambar wayar pelompat

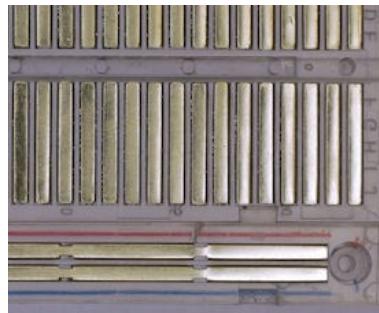
# UNIT 2.3

## Papan Reka

- Papan reka (breadboard) digunakan untuk menguji dan menghasilkan litar prototaip tanpa pateri.
- Lubang-lubang mempunyai sistem penyambungan seperti di rajah 2.3.(a). Lubang-lubang pada papan reka bersambung sepanjang garis merah. Komponen-komponen atau wayar akan bersambung jika berada pada garis yang sama.



Rajah 2.3.(a): Garis merah yang bersambung menunjukkan baris papan reka yang bersambung



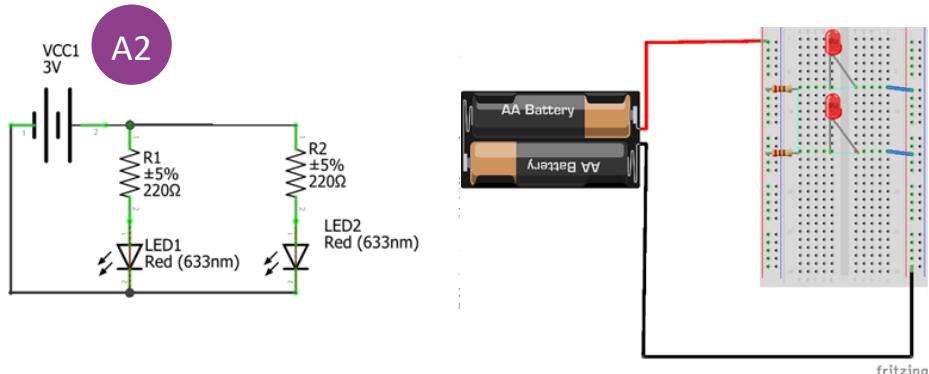
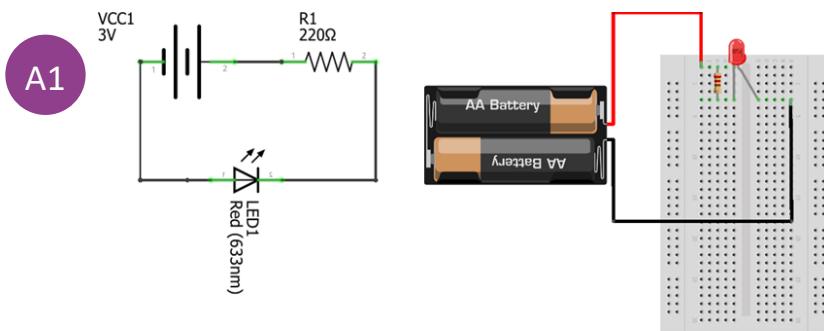
Rajah 2.3 (b): Bahagian dalam papan reka-perhati bahawa lajur besi yang menyambungkan litar

# UNIT 2.3

## Penyambungan litar output pada papan reka

### Penyambungan LED

Berikut adalah contoh untuk menyambungkan litar siri dan litar selari ringkas dengan menggunakan bateri 3V dan juga papan reka. Gambar rajah menunjukkan rajah skematik dan juga sambungan pada papan reka

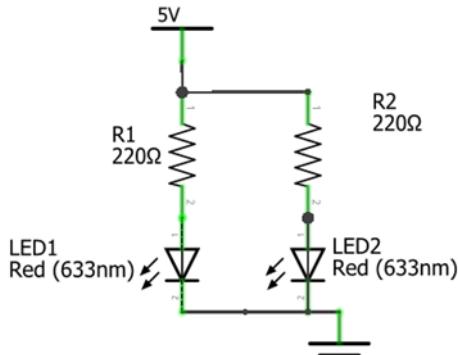
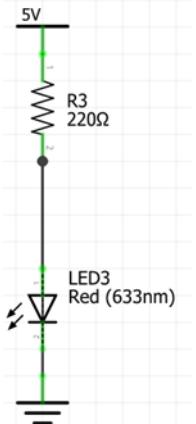


Rajah 2.2.(a): Litar A1 dan A2  
 (Atas) Litar siri  
 (Bawah) Litar selari

# UNIT 2.3

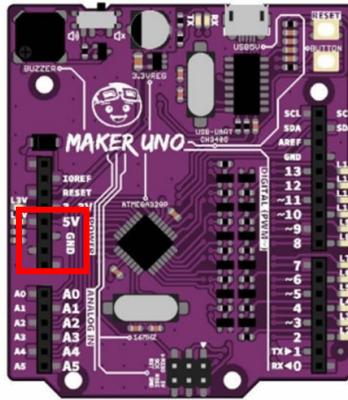
Bayangkan begini, sambungan bateri digantikan dengan sumber arus 5v.

B



fritzing

Arduino atau Maker UNO juga boleh membekalkan sumber voltan 5v seperti bateri dengan pin 5v dan juga pin GND (bumi).

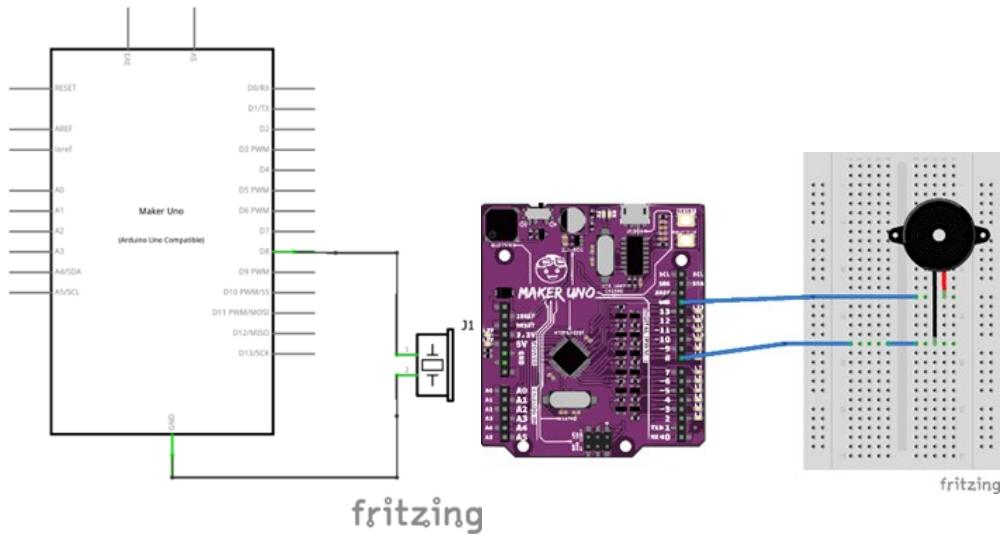


# UNIT 2.3

## Pembaz

Pembaz harus disambungkan mengikut kekutuhan seperti LED (merah ke pin dan hitam ke bumi). Berikut adalah contoh penyambungan pembaz pada Arduino. Untuk model Maker UNO, pembaz bina dalam telah disediakan pada pin 8. Pembaz tersebut perlu diaktifkan dengan menolak suis slaid sebelum digunakan.

Part1



# UNIT 2.3

Atur cara untuk pembaz

```

1 void setup() {
2   // put your setup c
3   pinMode(8, OUTPUT);
4 }
5
6 void loop() {
7   // put your main co
8   tone(8, 165, 500);
9   delay(1000);
10  tone(8, 175, 500);
11  delay(1000);
12  tone(8, 196, 500);
13  delay(1000);
14 }
15
16

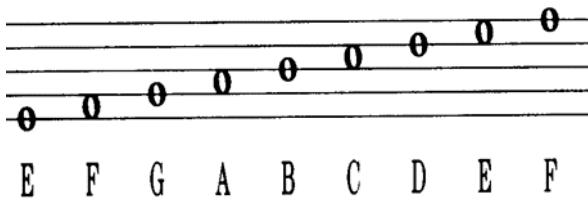
```

Baris	Pernyataan atur cara	Fungsi
1	tone (8, 165, 500)	Baris ini akan memainkan bunyi yang berfrekuensi 165Hz, iaitu not "E3" dalam muzik pada pin 8 selama 500 millisaat  <b>tone</b> merupakan fungsi khas untuk mengawal pembaz. Ia mengambil 3 maklumat iaitu nombor pin, frekuensi bunyi yang hendak dimainkan dan juga tempoh masa bunyi tersebut dimainkan

## UNIT 2.3

Tone mampu menghasilkan frekuensi yang bersamaan dengan not muzik. Berikut adalah senarai not muzik (pada oktaf 3) dan juga frekuensinya.

Not muzik	Frekuensi (Hz)
E	165
F	175
G	196
A	220
B	247
C	262
D	294
E	330
F	349

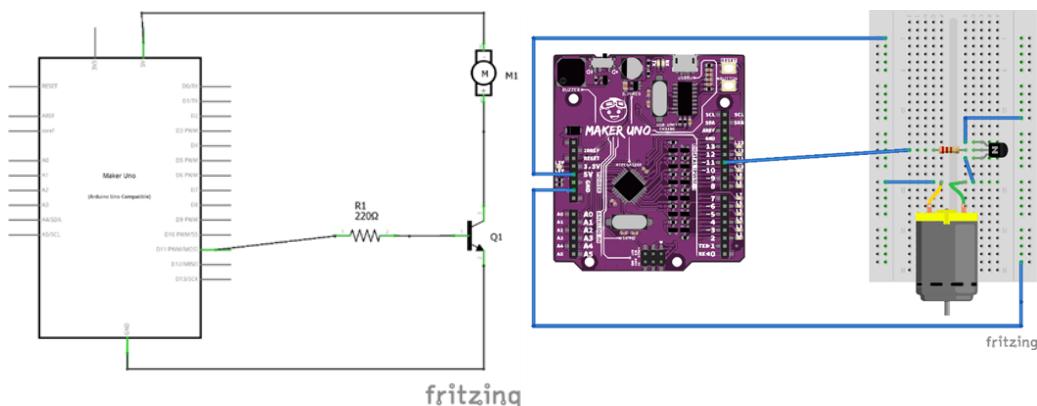


Senarai penuh semua nada muzik (dari oktaf 1 ke oktaf 8) boleh didapati di <https://www.arduino.cc/en/Tutorial/ToneMelody?from=Tutorial.Tone>

# UNIT 2.3

## Motor Arus Terus

Motor arus terus perlu disambung bersama dengan transistor – ini kerana motor memerlukan arus yang tinggi untuk berfungsi dan arus ini tidak dapat dibekalkan oleh pin *output* pada Arduino. Sambungan litarnya adalah seperti berikut



Untuk mengatasi masalah ini, kita memerlukan transistor yang akan berfungsi sebagai suis untuk mengawal motor. Untuk hidupkan motor, kita cuma perlu menghantar isyarat kepada pin 11 dan kemudian ia akan “membuka” transistor untuk membenarkan arus mengalir kepada motor.

Atur cara untuk menghidupkan motor adalah seperti berikut:

```
sketch_jan03a$ 
1 void setup() {
2   // put your setup code here,
3   pinMode(11, OUTPUT);
4 }
5
6 void loop() {
7   // put your main code here,
8   digitalWrite(11, HIGH);
9 }
```

## UNIT 2.4

# SIMULASI LITAR OUTPUT

### Objektif Pembelajaran

Dalam unit ini, murid akan dapat membuat simulasi litar output pada perisian khas

### Kriteria Kejayaan:

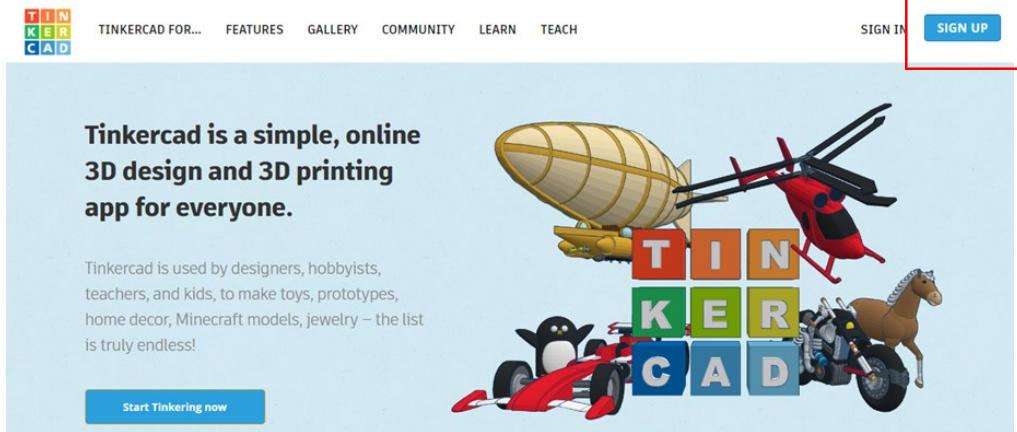
Murid dapat menghasilkan simulasi bagi sekurang-kurangnya 1 litar bersama dengan simulasi atur cara

Untuk menghasilkan simulasi litar *output*, kita boleh menggunakan laman web Tinkerlab secara percuma.

Laman web tinkerlab: [www.tinkerlab.com](http://www.tinkerlab.com)

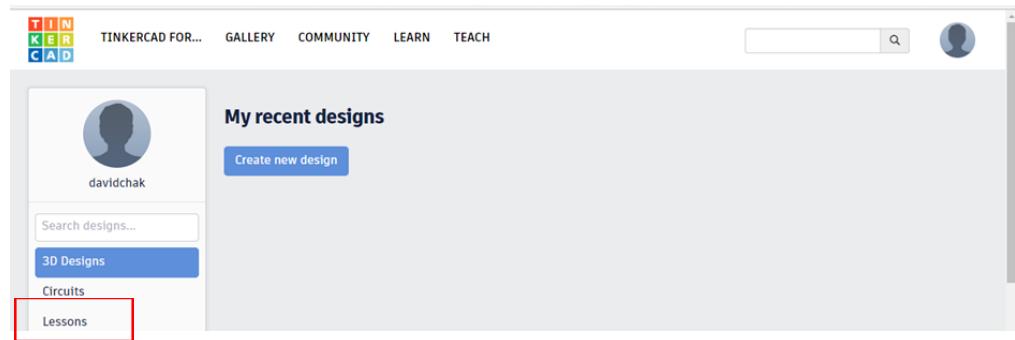
### Langkah-langkah untuk mengakses panduan penggunaan Tinkerlab

- 1) Daftar akaun pada tinkerlab

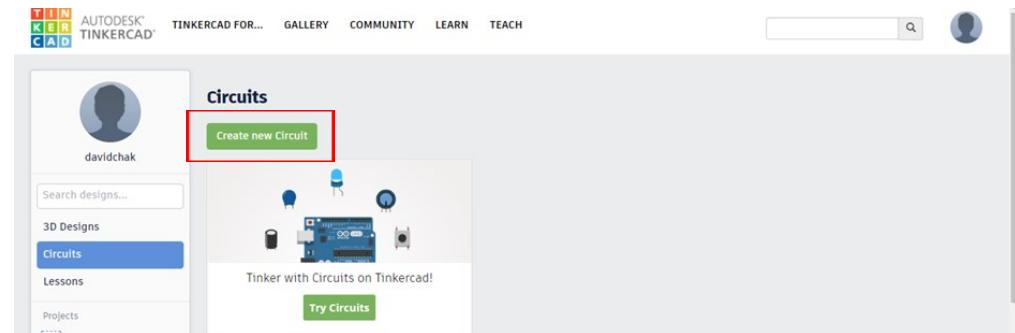


# UNIT 2.4

2) Log masuk akaun dan tekan “*Circuits*”



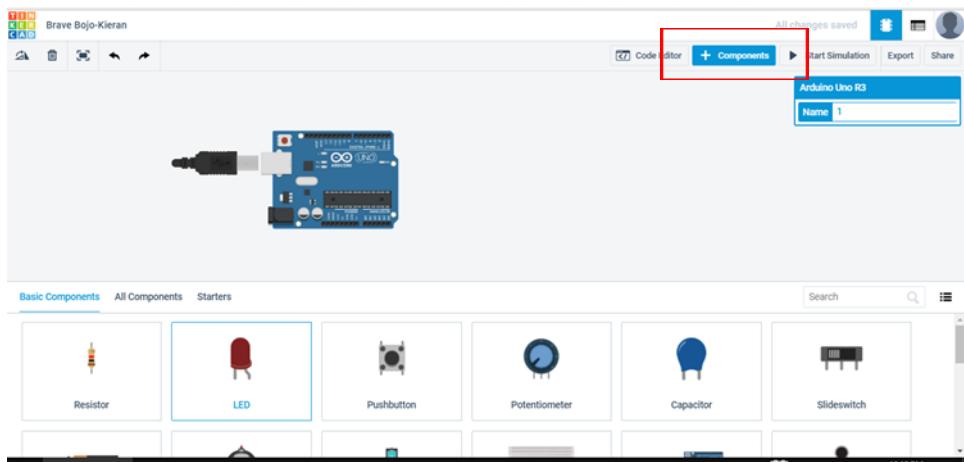
3) Tekan “*Create New Circuit*”



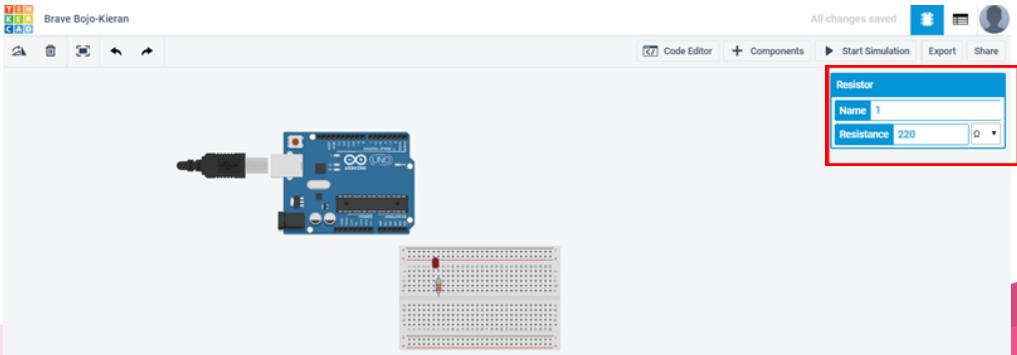
# UNIT 2.4

## Penghasilan simulasi litar 1 LED yang bersambung kepada pin 5

- 1) Tambahkan peranti LED, perintang, Arduino dan juga papan reka dengan butang “+ Components”

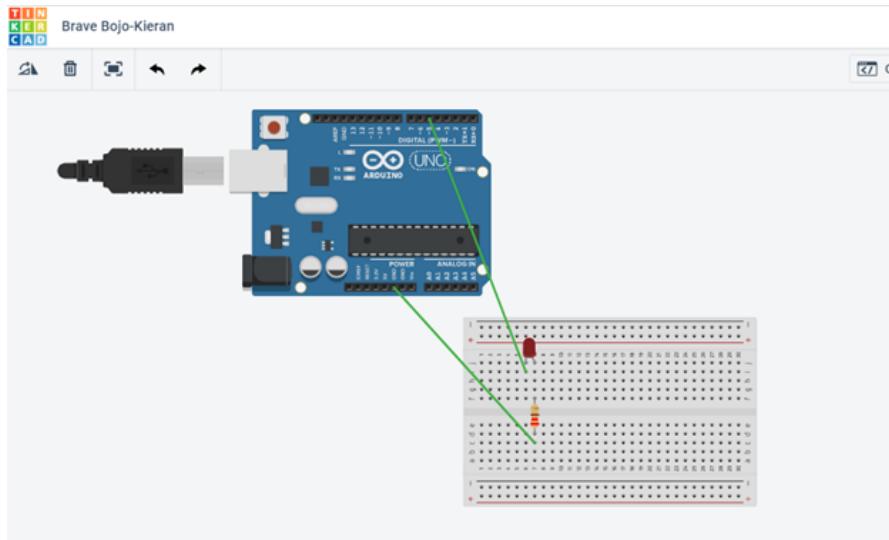


- 2) Pastikan nilai rintangan bagi perintang adalah betul dengan menekan sekali pada perintang dan mengubah nilai rintangannya

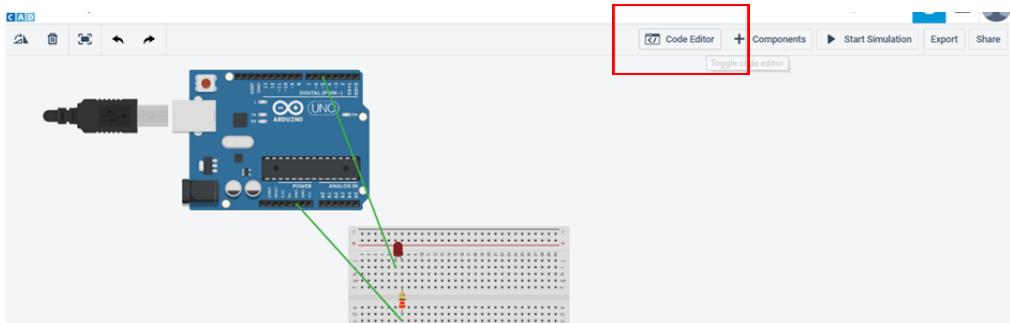


# UNIT 2.4

3) Sambungkan peranti-peranti ke atas papan reka dengan kaedah “drag and drop”

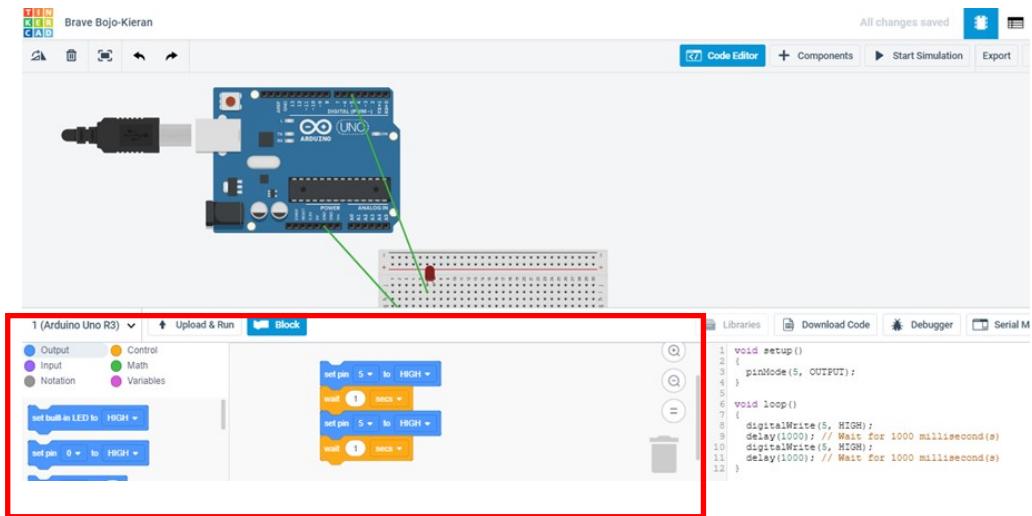


4) Tuliskan arur cara di bahagian “Code Editor”



# UNIT 2.4

- 5) "Drag and Drop" blok yang berkenaan untuk hasilkan atur cara bagi membolehkan LED berkelip di pin 5.



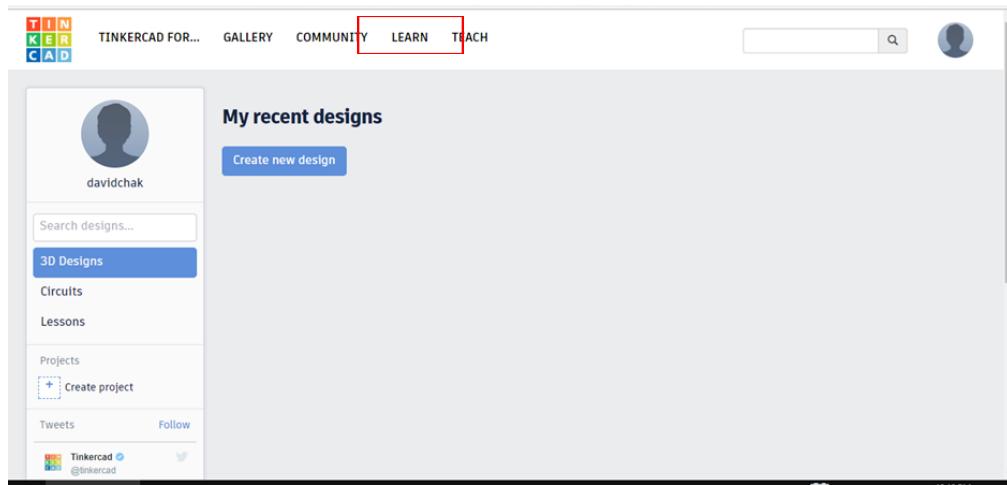
- 6) Setelah atur cara telah siap disusun, tekan *Upload and Run*

Cuba hasilkan simulasi litar *output* untuk lebih daripada 1 LED, pembaz ataupun motor. Untuk panduan lanjut dan lebih teliti boleh merujuk kepada panduan video ataupun panduan langkah-demi-langkah.

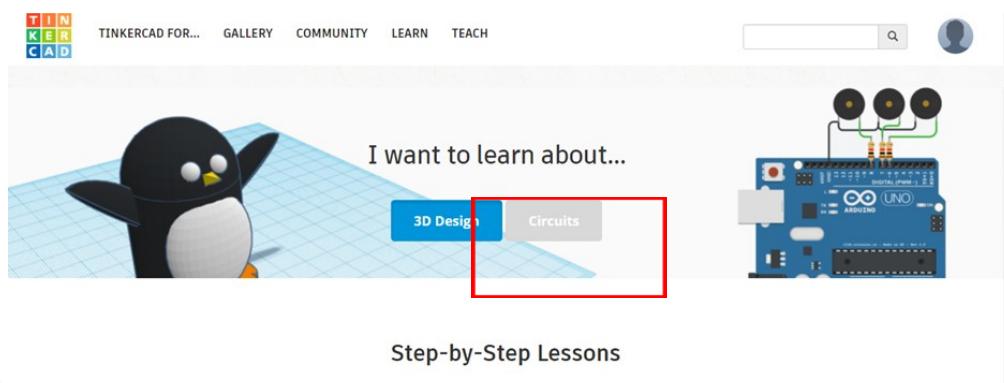
# UNIT 2.4

Langkah-langkah untuk mengakses panduan penggunaan Tinkerlab

- 1) Log masuk dalam akaun anda dan tekan "LEARN"



- 2) Pilih "Circuits"



# UNIT 2.4

3) Anda boleh mengikuti *step-by-step lesson* untuk belajar cara menggunakan Tinkerlab ataupun teori-teori tentang Arduino dan komponen-komponen elektronik



3) Anda juga boleh belajar melalui *video tutorial*.



Circuits Keyboard Shortcuts			
<b>General shortcuts</b>		<b>Moving object(s)</b>	
CTRL + C	Copy object(s)	SHIFT + M / M	Move one unit
CTRL + X	Cut object(s)	SHIFT + M - M / M - M	Move 1/10 unit
CTRL + V	Paste object(s)	M	Rotate Clockwise
CTRL + Z	Undo action(s)	SHIFT + M	Rotate Counterclockwise
CTRL + Y	Re-do action(s)		
CTRL + D	Deselect		
CTRL + S	Invert selection	Keyboard + mouse shortcuts (press and hold keyboard button, then move mouse)	SELECT + left mouse button
			Select multiple object(s)

# UNIT 3

# INPUT

## STANDARD PEMBELAJARAN

- 2.4.4 Membina litar simulasi yang berfungsi dengan perisian khas
- 2.4.5 Membuat penyambungan litar *input* dan *output* kepada mikropengawal
- 2.4.6 Menulis pengaturcaraan mudah berdasarkan penyambungan litar *input* dan *output*

## STANDARD PENILAIAN

TAHAP 4 Menguji kefungsian litar yang menggunakan mikropengawal (microcontroller).

## SUB-UNIT

UNIT 3.1	Pengenalan kepada pengaturcaraan litar input	SP: 2.4.6
UNIT 3.2	Jenis peranti input	
UNIT 3.3	Pengenalan kepada penyambungan litar input	SP: 2.4.5
UNIT 3.4	Simulasi litar input	SP: 2.4.5

## CADANGAN PEMBAHAGIAN MASA PDPC

UNIT 3.1	60 MINIT
UNIT 3.2	
UNIT 3.3	30 MINIT
UNIT 3.4	30 MINIT

## PERSEDIAAN

- 1) Pastikan ada set Arduino / Maker UNO yang mencukupi untuk murid
- 2) Unit 3.1 memerlukan komputer atau telefon pintar untuk pengaturcaraan. (Rujuk Lampiran 1)
- 3) Unit 3.4 memerlukan komputer dan Internet yang berfungsi
- 4) Boleh digunakan bersama dengan modul murid

# UNIT 3.1

# PENGENALAN KEPADA

# PENGATURCARAAN LITAR INPUT

## Objektif Pembelajaran

Dalam unit ini, murid akan dapat menulis aturcara mudah bagi mengawal litar input

## Kriteria Kejayaan:

Murid dapat menulis aturcara untuk membaca dan memaparkan nilai dari litar input

Mikropengawal dapat menerima isyarat melalui pin pada mikropengawal. Penderia (sensor) yang bersambung pada mikropengawal boleh menghantar isyarat digital ataupun isyarat analog.

Isyarat digital mempunyai dua jenis nilai iaitu sama ada 1 (*On*) atau 0 (*Off*). Semua pin bernombor (3-13) boleh menerima isyarat digital melalui kod “[digitalRead](#)”. Contoh peranti *input* yang menggunakan isyarat digital adalah peranti yang ada dua keadaan, seperti suis (suis ditekan / tidak ditekan), penderia hujan (ada air / tiada air) dan sebagainya.

Isyarat analog pula mengandungi lebih daripada dua nilai, iaitu merangkumi nilai 0 ke 1023. Isyarat analog hanya boleh dibaca melalui pin analog iaitu pin A0 ke A5 melalui kod “[analogRead](#)”. Contoh peranti *input* yang menggunakan isyarat analog adalah peranti yang mengesan lebih daripada dua keadaan, seperti perintang boleh laras (nilai berubah ikut pusingan knob), penderia cahaya (sangat gelap ke sangat terang), mikrofon dan sebagainya.

Sebelum kita menggunakan sebarang peranti *input*, kita perlu tetapkan pin tersebut sebagai **INPUT** di bahagian setup seperti begini:

sketch\_jan07a §

```
1 void setup() {  
2     // put your setup code here  
3     pinMode(4, INPUT);  
4 }  
5
```

\*Gantikan 4 dengan nilai pin yang digunakan dalam litar

# UNIT 3.1

Untuk Maker UNO, terdapat satu suis yang dibina-dalam pada pin 2. Untuk menggunakan pin tersebut, kita perlu tetapkannya sebagai `INPUT_PULLUP` seperti begini:

```
sketch_jan07a $ 
1 void setup() {
2     // put your setup code here,
3     pinMode(2, INPUT_PULLUP);
4 }
```

Untuk menggunakan nilai yang dibaca, kita perlu menulis atur cara dengan `analogRead` ataupun `digitalRead` dan simpankan nilai tersebut dalam satu pemboleh ubah seperti begini:

```
6 void loop() {
7     // put your main code here,
8     int x = digitalRead(2);
9     int y = analogRead(A2);
10 }
```

Baris 8 membaca nilai isyarat digital dari pin 2 dan simpankan dalam pemboleh ubah x manakala baris 9 membaca nilai isyarat analog dari pin A2 dan simpankan dalam pemboleh ubah y.

Kita juga boleh membuka komunikasi serial dengan komputer supaya kita boleh dapatkan memaparkan nilai untuk tujuan penghasilan projek. Untuk berbuat demikian, kita perlu membuka saluran di antara Arduino dengan komputer pada bahagian setup seperti berikut:

```
1 void setup() {
2     // put your setup code
3     pinMode(A0, INPUT);
4     Serial.begin(9600);
5 }
```

9600 adalah baudrate penghantaran data dari Arduino ke komputer. Semakin tinggi nilai, semakin banyak data yang boleh dihantar dari Arduino ke komputer. Baudrate yang biasa digunakan adalah 9600

# UNIT 3.1

Seterusnya, kita boleh paparkan nilai yang dibaca pada komputer dengan menggunakan Serial.println (l adalah L huruf kecil)

```
1 void setup() {  
2  
3     pinMode(A0, INPUT);  
4     Serial.begin(9600);  
5 }  
6  
7 void loop() {  
8  
9     int x = analogRead(A2);  
10    Serial.println(x);  
11 }
```

Setelah kod dimuat naik ke Arduino, kita boleh bukakan Serial Monitor (CTRL+SHIFT+M) ataupun menekan butang berikut untuk melihat nilai yang dibaca oleh Arduino.



# UNIT 3.1

Contoh kod yang lengkap di bawah adalah untuk membaca suis yang bersambung pada pin 2 (untuk Maker UNO), dan memaparkan nilai tersebut. Contoh ini tidak memerlukan sambungan pada papan reka untuk Maker UNO.

Untuk Arduino lain, litar pada papan reka diperlukan, iaitu suis disambung pada pin 2 dan bahagian setup perlu ditukarkan kepada `pinMode(2, INPUT)`.

sketch\_jan07a §

```
1✉ void setup() {  
2    // put your setup code here,  
3    pinMode(2, INPUT_PULLUP);  
4    Serial.begin(9600);  
5}  
6  
7✉ void loop() {  
8    // put your main code here,  
9    int x = digitalRead(2);  
10   Serial.println(x);  
11}  
12  
13
```

## UNIT 3.2

# JENIS PERANTI INPUT

### Objektif Pembelajaran

Dalam unit ini, murid akan dapat mengenali pelbagai jenis peranti *input*

### Kriteria Kejayaan:

Murid dapat menyenaraikan nama dan fungsi pelbagai jenis peranti *input*

Peranti *input* akan membaca nilai yang dikesan dari keadaan persekitaran dan menghantar maklumat yang dibaca melalui pin pada Arduino. Semua pin boleh digunakan untuk membaca nilai isyarat digital melalui kod **digitalRead**.

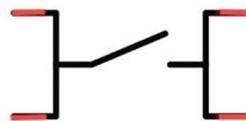
Isyarat analog hanya boleh dikesan dengan menggunakan pin analog iaitu pin A0 ke A5. Terdapat pelbagai peranti *input* yang boleh digunakan, seperti suis, perintang boleh laras, perintang peka cahaya ataupun penderia cahaya, penderia suhu, penderia manusia (PIR Sensor) dan sebagainya.

### Suis

- Terdapat pelbagai jenis suis yang wujud seperti mikrosuis, suis togel (toggle switch) dan lain-lain.
- Walaupun terdapat banyak jenis suis yang wujud di pasaran, semua suis berfungsi dengan prinsip yang sama, iaitu sama ada litar lengkap ataupun tidak.
- Untuk maklumat lebih lanjut tentang setiap jenis suis, boleh merujuk kepada laman web: <https://learn.sparkfun.com/tutorials/switch-basics> dan <https://electronicsclub.info/switches.htm>
- Simbol dan gambar untuk suis adalah seperti berikut:
- Suis Jenis tekan tutup (*Pushbutton*)
  - ⇒ Suis ini akan menukar keadaan ke “ON” selagi ia ditekan. Apabila dilepaskan, ia akan kembali ke keadaan “OFF”.



Rajah 3.1.(a): Gambar suis tekan tutup



Rajah 3.1.(b): Simbol skematik untuk suis tekan tutup

# UNIT 3.2

## Suis

- Suis Togel (*Toggle Switch*)  
⇒ Suis Jenis SPST – *Single Pole, Single Throw*



SPST

Suis jenis SPST mempunyai dua terminal, dan sambungannya boleh tentukan sama ada litar itu lengkap ataupun putus. Sambungan suis jenis SPST memerlukan litar pull-up / pull-down untuk dapatkan bacaan yang lebih jitu. Ia akan kekal dalam bentuk "ON" atau "OFF" bergantung kepada lokasi butang.

⇒ Suis jenis SPDT – *Single Pole, Double Throw*



SPDT

Suis jenis SPDT mempunyai 3 terminal, iaitu terminal umum (*common*), terminal NO (*Normally Open*) dan NC (*Normally Close*). Sambungan suis jenis SPDT tidak memerlukan litar *pull-up/pull-down*. Ia akan kekal dalam bentuk ON atau OFF bergantung kepada lokasi butang.

Rajah 3.1.(b): Simbol skematik untuk suis tekan tutup

# UNIT 3.2

## Suis

- Mikrosuis (*Microswitch*)



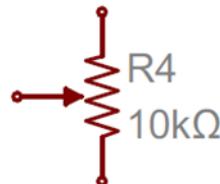
Mikrosuis merupakan suis yang bertindak dengan pergerakan yang kecil ataupun tekanan ringan pada suis. Contoh mikrosuis yang wujud dalam kehidupan harian kita adalah suis lampu dalam peti sejuk. Apabila peti sejuk ditutup, lampu dalam peti sejuk akan padam.

## Perintang Boleh Laras / Meter Upaya (*Potentiometer*)

- Perintang boleh laras merupakan perintang yang mempunyai nilai rintangan yang tidak tetap. Nilai rintangannya dapat diubah.
- Nilai rintangan boleh diubah dengan memutar tombol pelaras.
- Perubahan rintangan tersebut boleh dibaca oleh Arduino.



Rajah 3.1.(c): Gambar Perintang Boleh Laras

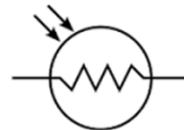


Rajah 3.1.(d): Simbol skematik untuk perintang boleh laras

# UNIT 3.2

## Perintang Peka Cahaya / Penderia Cahaya (*Light Dependent Resistor*)

- Rintangan peka cahaya berubah mengikut keterangan cahaya - semakin terang, semakin tinggi nilai yang dibaca oleh Arduino

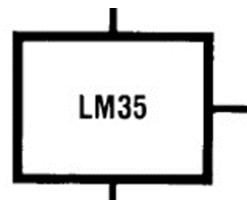


Rajah 3.1.(e): Gambar Perintang peka cahaya

Rajah 3.1.(f): Simbol skematic untuk perintang peka cahaya

## Penderia Suhu

- Terdapat pelbagai jenis penderia suhu yang terdapat di pasaran, penderia suhu yang paling biasa digunakan adalah model LM35.
- Hasil LM35 adalah dalam bentuk voltan yang dibaca, untuk dapatkan hasil dalam bentuk celsius, pengiraan yang lebih mendalam diperlukan.
- Langkah pengiraan boleh merujuk kepada laman web:  
<https://tutorial.cytron.io/2017/07/13/getting-started-temperature-sensor-celsius-sn-lm35dz/>



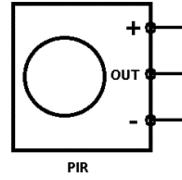
Rajah 3.1.(g): Gambar penderia suhu

Rajah 3.1.(h): Simbol skematic untuk penderia suhu.  
Simbol ini adalah contoh symbol IC (*Integrated circuit*)

# UNIT 3.2

## Penderia Pergerakkan Pasif (*PIR Sensor*)

- Pergerakan manusia boleh dikesan dengan penderia PIR (Passive Infrared) yang menggunakan cahaya inframerah.



Rajah 3.1.(f): Simbol skematik untuk perintang PIR

Rajah 3.1.(e): Gambar penderia PIR

## UNIT 3.3

# PENGENALAN KEPADA PENYAMBUNGAN LITAR INPUT

### Objektif Pembelajaran

Dalam unit ini, murid akan dapat membaca rajah skematik dan menyambungkan litar berdasarkan rajah skematik

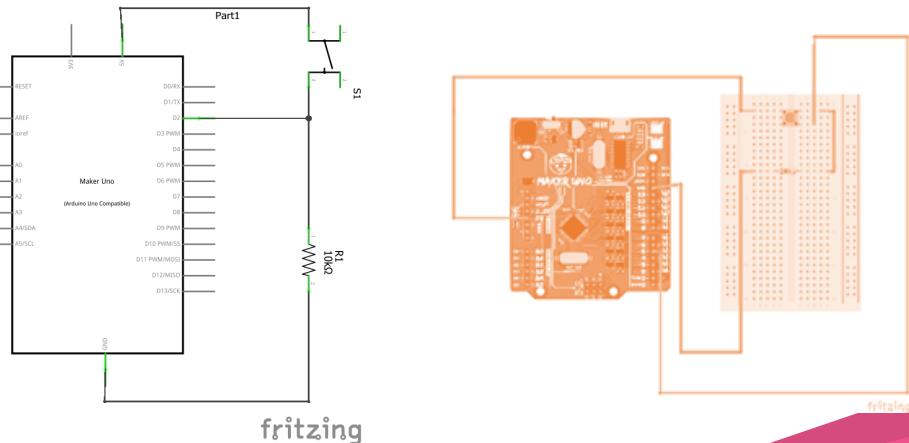
### Kriteria Kejayaan:

Murid dapat menghasilkan sekurang-kurangnya 1 litar berdasarkan rajah skematik yang diberi

Seperti litar *output*, penyambungan litar juga akan menggunakan wayar pelompat dan papan reka.

### Suis Jenis Tekan Tutup

- Maker UNO mempunyai satu suis tekan tutup pada pin 2. Untuk gunakan suis tersebut, kita perlu tetapkan pinMode sebagai **INPUT\_PULLUP** pada bahagian setup.
- Sambungan untuk suis tekan tutup pada papan reka adalah seperti berikut:



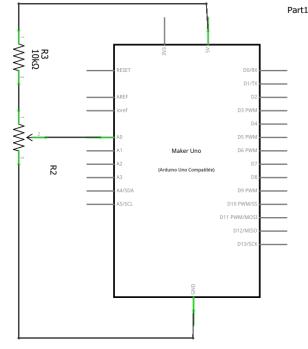
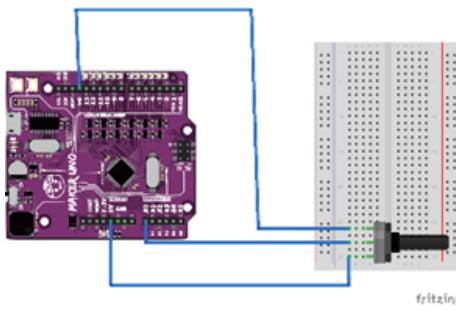
# UNIT 3.3

## Suis Jenis Tekan Tutup

- Suis merupakan peranti jenis digital, jadi kita perlu gunakan kod `digitalRead` untuk bacakan nilai suis.
- Perintang 10k diperlukan untuk wujudkan litar *PULL DOWN* iaitu ia memastikan bahawa nilai yang dibaca adalah tepat dengan menarik (*PULL*) semua cas elektrik ke bumi apabila suis tidak ditekan.
- Maklumat lebih lanjut tentang litar *PULL DOWN* boleh didapatkan pada laman web: <https://learn.sparkfun.com/tutorials/pull-up-resistors> dan [http://www.resistorguide.com/pull-up-resistor\\_pull-down-resistor/](http://www.resistorguide.com/pull-up-resistor_pull-down-resistor/)

## Perintang Boleh Laras

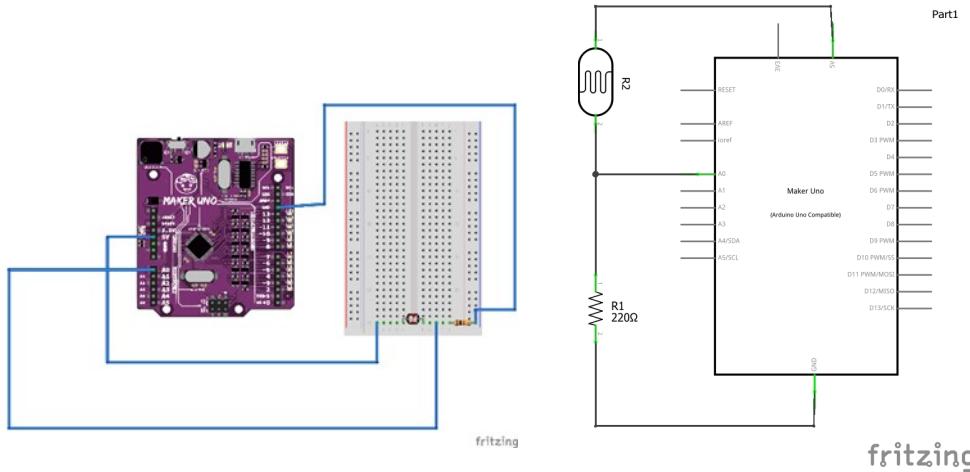
- Sambungan perintang boleh laras tidak mempunyai keikutinan kerana ia adalah perintang secara amnya.
- Pin tamatan kiri dan kanan perintang boleh laras disambung kepada 5v dan bumi manakala pin tengah perlu disambungkan kepada pin yang hendak digunakan.
- Perintang boleh laras merupakan peranti *input* jenis analog, jadi kita perlu gunakan pin A0 ke A5 dan gunakan arahan kod `analogRead` untuk bacakan nilai perintang boleh laras.



# UNIT 3.3

## Perintang Peka Cahaya

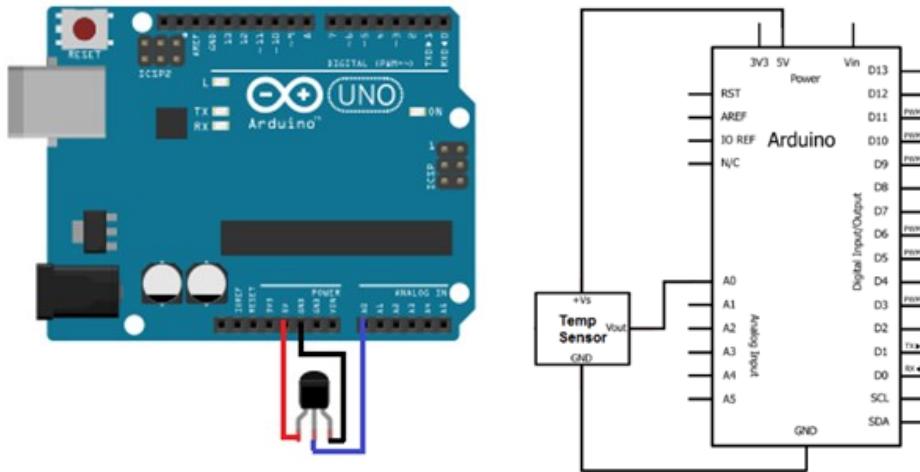
- Seperti perintang boleh laras, perintang peka cahaya juga tidak mempunyai kekutuhan.
- Seperti sambungan suis juga, perintang peka cahaya memerlukan litar *PULL DOWN* untuk dapatkan bacaan yang jitu.
- Perintang peka cahaya merupakan peranti input jenis analog, jadi kita perlu gunakan pin A0 – A5 dan gunakan arahan kod **analogRead** untuk membaca nilai perintang



# UNIT 3.3

## Penderia Suhu

- Sambungan penderia suhu LM35 adalah seperti sambungan perintang boleh laras di mana antara 3 tamatan, tamatan tengah disambungkan kepada pin. Tamatan kiri dan kanan disambungkan kepada 5v dan juga bumi
- Penderia suhu merupakan peranti *input* jenis analog, jadi kita perlu gunakan pin A0 - A5 dan gunakan arahan kod **analogRead** untuk bacakan nilai penderia suhu



## UNIT 3.4

# SIMULASI LITAR INPUT

### Objektif Pembelajaran

Dalam unit ini, murid akan dapat membuat simulasi litar input pada perisian khas

### Kriteria Kejayaan:

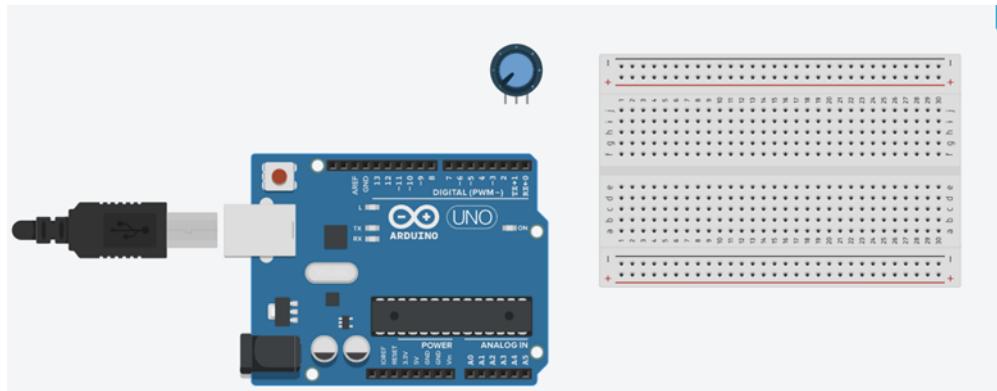
Murid dapat menghasilkan simulasi bagi sekurang-kurangnya 1 litar bersama dengan simulasi arut cara

Untuk menghasilkan simulasi litar *input*, kita boleh menggunakan laman web Tinkercad secara percuma.

Laman web tinkercad: [www.tinkercad.com](http://www.tinkercad.com)

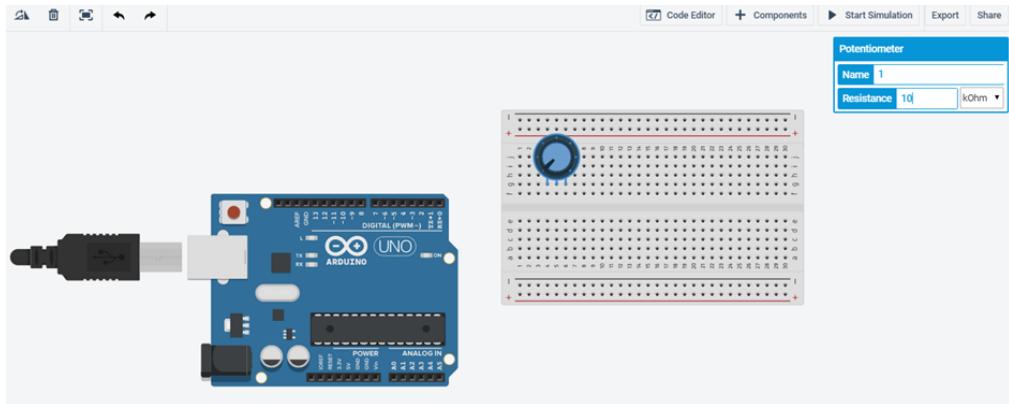
### Penghasilan simulasi litar perintang boleh laras

- 1) Tambahkan peranti-peranti perintang boleh laras, Arduino dan juga papan reka dengan butang “+ Components”

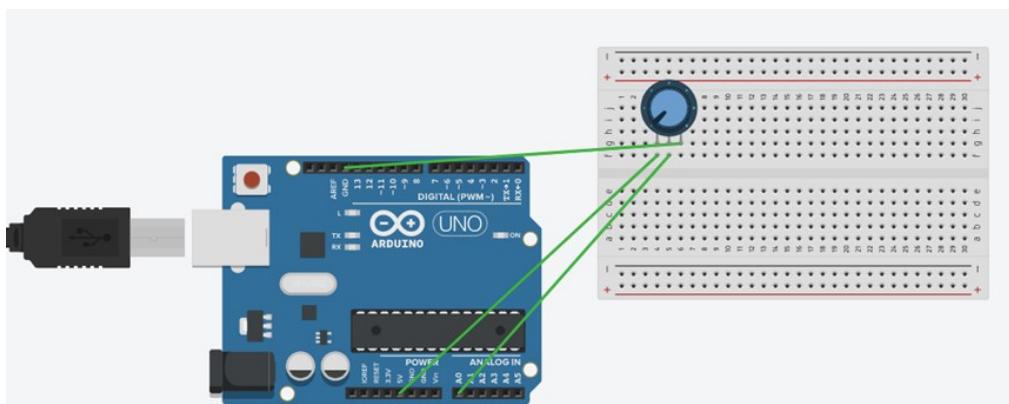


# UNIT 3.4

2) Pastikan nilai perintang boleh laras ditetapkan dengan betul dengan memilih perintang boleh laras dan menukar nilai rintangannya kepada 10k ataupun 100k.

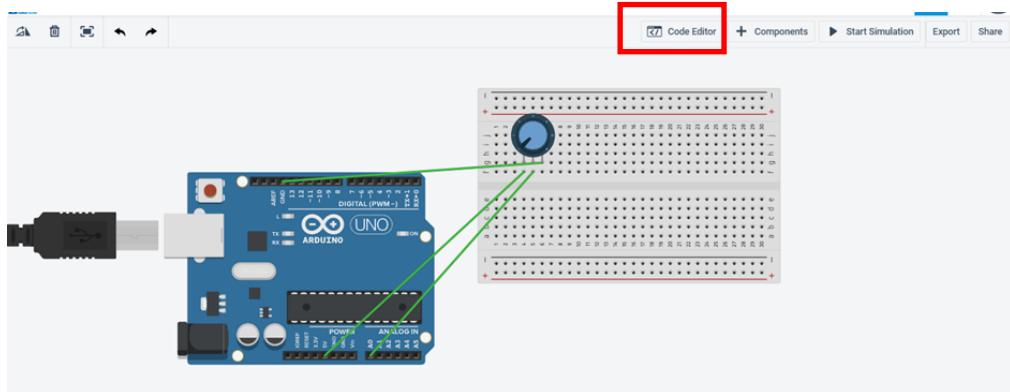


3) Sambungkan peranti-peranti ke atas papan reka dengan kaedah “drag and drop”



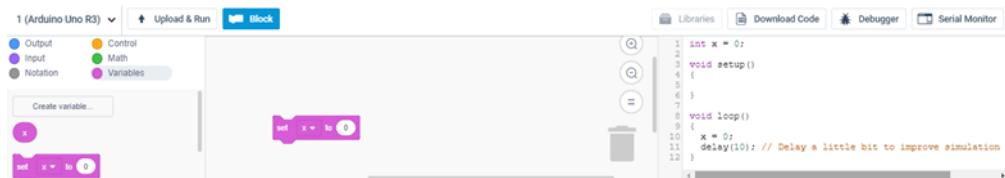
# UNIT 3.4

4) Tuliskan aturcara di bahagian “Code Editor”



5) “Drag and Drop” blok yang berkenaan untuk hasilkan atur cara bagi membaca nilai perintang boleh laras yang bersambung pada pin A0:

a) Hasilkan pemboleh ubah baru



# UNIT 3.4

b) Tetapkan nilai pemboleh ubah tersebut kepada nilai bacaan pin A0.

The screenshot shows the Arduino IDE interface. On the left is the Scratch-style code editor with blocks for Output, Input, Control, Math, and Variables. On the right is the text-based code editor with the following code:

```

1 int x = 0;
2
3 void setup()
4 {
5   pinMode(A0, INPUT);
6 }
7
8 void loop()
9 {
10   x = analogRead(A0);
11   delay(10); // Delay a little bit to improve simulation;
12 }

```

c) Paparkan nilai bacaan dengan komunikasi Serial (gabungan blok Output dan blok Variables)

The screenshot shows the Arduino IDE interface. On the left is the Scratch-style code editor with blocks for Output, Input, Control, Math, and Variables. On the right is the text-based code editor with the following code:

```

1 int x = 0;
2
3 void setup()
4 {
5   pinMode(A0, INPUT);
6   Serial.begin(9600);
7 }
8
9 void loop()
10 {
11   x = analogRead(A0);
12   Serial.println(x);
13 }

```

6) Setelah atur cara telah siap disusun, tekan *Upload and Run*.

The screenshot shows the Arduino IDE interface. A red box highlights the 'Upload & Run' button in the Scratch editor toolbar. On the left is the Scratch-style code editor with blocks for Output, Input, Control, Math, and Variables. On the right is the text-based code editor with the following code:

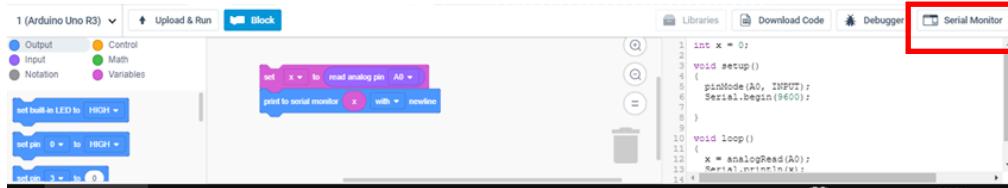
```

1 int x = 0;
2
3 void setup()
4 {
5   pinMode(A0, INPUT);
6   Serial.begin(9600);
7 }
8
9 void loop()
10 {
11   x = analogRead(A0);
12   Serial.println(x);
13 }

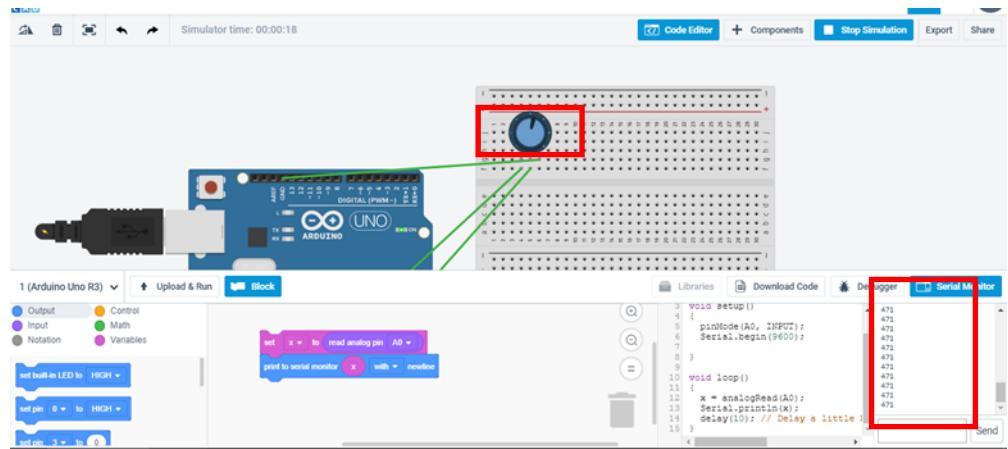
```

# UNIT 3.4

7) "Serial Monitor" boleh disimulasikan dengan butang "Serial Monitor".



8) Apabila kita putarkan tombol pelaras, paparan nilai di "Serial Monitor" juga turut berubah.



# UNIT 4

# GABUNGAN INPUT/OUTPUT

## STANDARD PEMBELAJARAN

- 2.4.4 Membina litar simulasi yang berfungsi dengan perisian khas
- 2.4.5 Membuat penyambungan litar *input* dan *output* kepada mikropengawal
- 2.4.6 Menulis pengaturcaraan mudah berdasarkan penyambungan litar *input* dan *output*
- 2.4.7 Membuat pengujian dan penilaian kefungsian litar elektronik
- 2.4.8 Mencadangkan penambahbaikan ke atas reka bentuk litar elektronik

## STANDARD PENILAIAN

- TAHAP 4 Menguji kefungsian litar yang menggunakan mikropengawal (microcontroller).
- TAHAP 5 Membuat justifikasi kawalan atur cara *input* dan *output* bagi menyelesaikan masalah
- TAHAP 6 Membina litar elektronik kawalan mikropengawal (microcontroller) yang berfungsi

## SUB-UNIT

- UNIT 4.1 Pengenalan kepada pengaturcaraan struktur kawalan pilihan SP: 2.4.6
- UNIT 4.2 Gabungan litar *input* dan *output* SP: 2.4.7
- UNIT 4.3 Simulasi gabungan litar *input* dan *output* SP 2.4.5
- UNIT 4.4 Penghasilan projek SP: 2.4.7 / 2.4.8

## CADANGAN PEMBAHAGIAN MASA PDPC

UNIT 4.1	60 MINIT
UNIT 4.2	
UNIT 4.3	30 MINIT
UNIT 4.4	90 MINIT

## PERSEDIAAN

- 1) Pastikan ada set Arduino / Maker UNO yang mencukupi untuk murid
- 2) Unit 4.1 memerlukan komputer atau telefon pintar untuk pengaturcaraan. (Rujuk Lampiran 1)
- 3) Unit 4.3 memerlukan komputer dan Internet yang berfungsi
- 4) Boleh digunakan beserta dengan modul murid

## UNIT 4.1

# PENGENALAN KEPADA PENGATURCARAAN STRUKTUR KAWALAN PILIHAN

### Objektif Pembelajaran

Dalam unit ini, murid akan dapat menulis arah cara yang mengeluarkan *output* berlainan berdasarkan *input* yang diterima

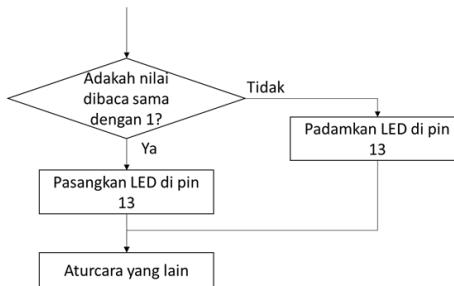
### Kriteria Kejayaan:

Murid dapat menulis arah cara untuk memasang LED yang terbina dalam dengan menekan suis yang bersambung atau terbina dalam Arduino

Kita boleh gabungkan *input* dan *output* untuk menghasilkan sesuatu projek yang boleh berinteraksi dengan persekitaran.

Imbas kembali bahawa mikropengawal berfungsi sebagai otak untuk mengeluarkan respons (*output*) yang sewajarnya berdasarkan maklumat (*input*) yang diterima.

Dengan menggunakan peranti *input*, kita boleh menggunakan nilai yang dibaca dan bandingkan dengan nilai yang kita kehendaki. Sekiranya nilai yang dibaca sepadan dengan corak yang diperlukan, mikropengawal boleh menghasilkan output yang dikehendaki. Ini disebut sebagai struktur kawalan pilihan. Perwakilan dalam carta alir adalah seperti begini:



Rajah 4.1.(a): Carta alir untuk struktur kawalan pilihan

# UNIT 4.1

Aturcara untuk carta alir di rajah 4.1(a) adalah seperti berikut:

```

6 void loop() {
7     // put your main code here,
8     int x = digitalRead(2);
9     if (x == 1) {
10         digitalWrite(13, HIGH);
11     } else {
12         digitalWrite(13, LOW);
13     }
14 }
```

Operator perbandingan digunakan untuk membandingkan dua nilai dan akan hasilkan nilai benar jika perbandingan tersebut adalah sahih. Terdapat 6 jenis operator perbandingan dalam pengaturcaraan Arduino:

Operator Perbandingan	Maksud	Contoh
<code>==</code>	Sama dengan	<pre>x = digitalRead(5); if (x == 1){     digitalWrite(13, HIGH); } else {     digitalWrite(13, LOW); }</pre> <p>Baca nilai x (suis bersambung pada pin 5). Jika x adalah 1, iaitu suis ditekan, maka isyarat digital dihantar untuk menyalakan LED pada pin 13. Jika tidak, hantar isyarat digital untuk memadamkan LED pada pin 13.</p>

# UNIT 4.1

Operator Perbandingan	Maksud	Contoh
<b>!=</b>	Tidak sama dengan	<pre>x = digitalRead(5); if (x != 1){     digitalWrite(13, HIGH); } else {     digitalWrite(13, LOW); }</pre> <p>Baca nilai x (suis bersambung pada pin 5). Jika x tidak sama dengan 1, iaitu suis tidak ditekan, maka isyarat digital dihantar untuk menyalakan LED pada pin 13. Jika suis ditekan (nilai adalah 1), isyarat digital dihantar untuk memadamkan LED pada pin 13.</p>
<b>&gt;=</b>	Lebih besar atau sama dengan	<pre>x = analogRead(A0); if (x &gt;= 100){     digitalWrite(13, HIGH); } else {     digitalWrite(13, LOW); }</pre> <p>Baca nilai x iaitu peranti <i>input</i> analog yang bersambung ke pin A0. Jika nilai yang dibaca adalah 100 atau lebih daripada 100, nyalakan LED pada pin 13. Kalau tidak, padamkan LED pada pin 13.</p>

# UNIT 4.1

Operator Perbandingan	Maksud	Contoh
>	Lebih besar daripada	<pre>x = analogRead(A0); if (x &gt; 100){     digitalWrite(13, HIGH); } else {     digitalWrite(13, LOW); }</pre> <p>Baca nilai x iaitu peranti <i>input</i> analog yang bersambung ke pin A0. Jika nilai yang dibaca adalah lebih daripada 100 (tidak termasuk nilai 100), nyalakan LED pada pin 13. Kalau tidak, padamkan LED pada pin 13.</p>
<=	Lebih kecil atau sama dengan	<pre>x = analogRead(A0); if (x &lt;= 100){     digitalWrite(13, HIGH); } else {     digitalWrite(13, LOW); }</pre> <p>Baca nilai x iaitu peranti <i>input</i> analog yang bersambung ke pin A0. Jika nilai yang dibaca adalah 100 atau kurang daripada 100, nyalakan LED pada pin 13. Kalau tidak, padamkan LED pada pin 13.</p>

# UNIT 4.1

Operator Perbandingan	Maksud	Contoh
<	Lebih kecil daripada	<pre>x = analogRead(A0); if (x &lt; 100){     digitalWrite(13, HIGH); } else {     digitalWrite(13, LOW); }</pre> <p>Baca nilai x iaitu peranti <i>input</i> analog yang bersambung ke pin A0. Jika nilai yang dibaca adalah kurang daripada 100 (tidak termasuk nilai 100), nyalakan LED pada pin 13. Kalau tidak, padamkan LED pada pin 13.</p>

Untuk dapatkan nilai di antara dua nilai, kita perlu gunakan operator logik “AND”.

Contohnya, nilai di antara 500 dan 700 boleh dikesan dengan atur cara berikut:

```

6 void loop() {
7     // put your main code here,
8     int x = analogRead(A2);
9     if (x > 500 && x < 700) {
10         digitalWrite(13, HIGH);
11     } else {
12         digitalWrite(13, LOW);
13     }
14 }
```

(Simbol untuk AND adalah &&)

# UNIT 4.1

Contoh 1: (Khas Untuk Maker UNO)

Contoh kod yang lengkap dibawah adalah membaca suis yang bersambung pada pin 2 (untuk Maker UNO), memaparkan nilai tersebut, dan jika suis ditekan, nyalakan LED pada pin 13. Jika suis tidak ditekan, padamkan LED tersebut.

(Bagi Arduino lain, sila sambungkan litar pada papan reka serta tukarkan `INPUT_PULLUP` ke `INPUT`)

```
sketch_jan07a§
1 void setup() {
2
3     pinMode(2, INPUT_PULLUP);
4     pinMode(13, OUTPUT);
5     Serial.begin(9600);
6 }
7
8 void loop() {
9
10    int x = digitalRead(2);
11    Serial.println(x);
12    if (x == 1){
13        digitalWrite(13, HIGH);
14    } else {
15        digitalWrite(13, LOW);
16    }
17 }
```

# UNIT 4.1

Contoh 2: (Perlukan sambungan pada papan reka)

Contoh ini membaca nilai pada perintang peka cahaya, memaparkan dan mengeluarkan *output* berlainan bergantung kepada nilai yang dibaca. Apabila keterangan persekitaran menjadi gelap (nilai dibaca sama dengan atau kurang daripada 400), LED pada pin 13 akan dinyalakan. Jika keadaan menjadi terang (nilai dibaca lebih daripada 400), LED pada pin 13 akan dipadamkan.

```
sketch_jan07a\$ 
1 void setup() {
2
3     pinMode(A0, INPUT);
4     pinMode(13, OUTPUT);
5     Serial.begin(9600);
6 }
7
8 void loop() {
9
10    int x = analogRead(A0);
11    Serial.println(x);
12    if(x <= 400){
13        digitalWrite(13, HIGH);
14    } else {
15        digitalWrite(13, LOW);
16    }
17 }
```

# UNIT 4.2

# GABUNGAN LITAR INPUT DAN OUTPUT

## Objektif Pembelajaran

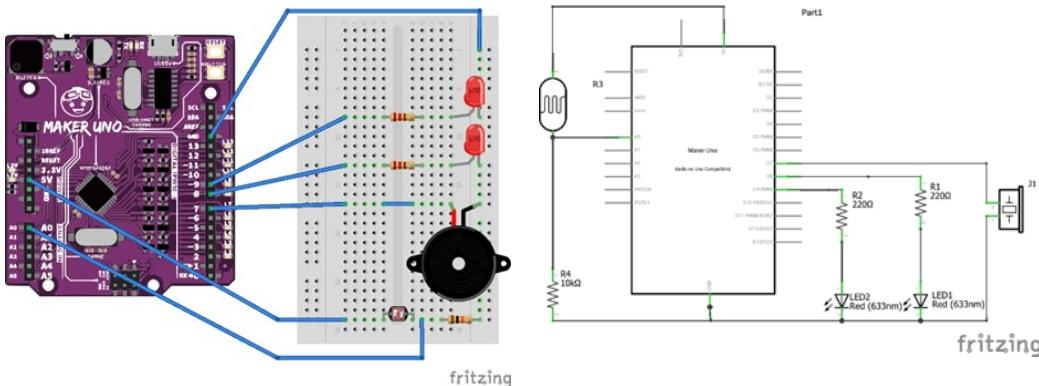
Dalam unit ini, murid akan dapat membaca rajah skematic dan menyambungkan litar berdasarkan rajah skematic

## Kriteria Kejayaan:

Murid dapat menghasilkan sekurang-kurangnya 1 litar berdasarkan rajah skematic yang diberi

Sambungan litar *input* dan *output* tidak banyak perbezaan dengan penyambungan litar *input* sahaja ataupun penyambungan litar *output* sahaja. Berikut adalah beberapa contoh yang melibatkan litar *input* dan *output*

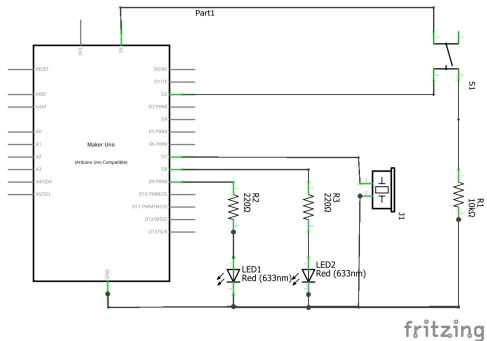
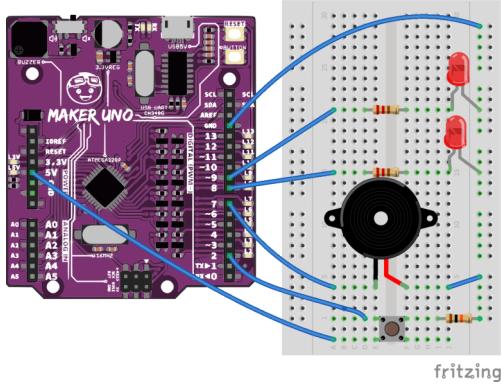
### Litar yang melibatkan perintang peka cahaya, pembaz dan LED



Dengan sambungan ini, kita boleh menghasilkan satu kotak muzik ringkas. Apabila kotak dibuka (kecerahan cahaya meningkat), mikropengawal akan memainkan muzik dan menyalaikan LED.

## UNIT 4.2

Litar yang melibatkan suis, pembaz dan LED



Dengan sambungan ini, kita boleh hasilkan satu kotak muzik yang diaktifkan oleh suis. Apabila suis ditekan, mikropengawal akan mengeluarkan muzik dan menyalakan LED.

## UNIT 4.3

# SIMULASI INPUT DAN OUTPUT

### Objektif Pembelajaran

Dalam unit ini, murid akan dapat membuat simulan litar *input* dan *output* pada perisian khas

### Kriteria Kejayaan:

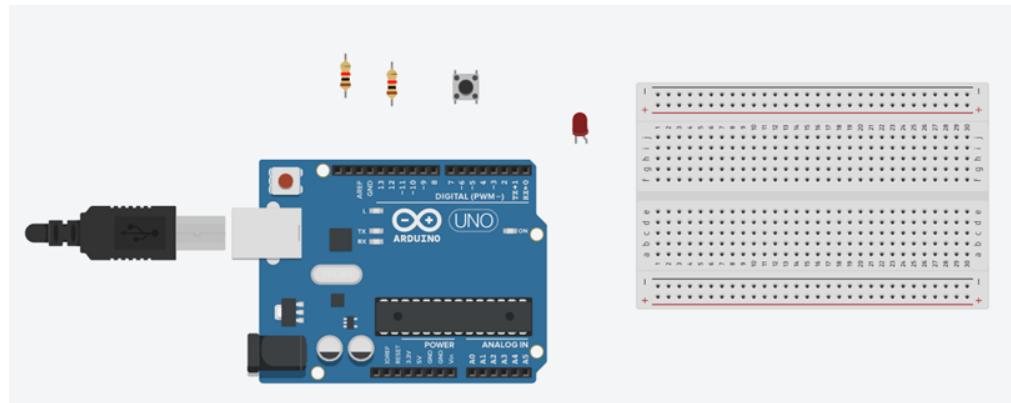
Murid dapat menghasilkan simulan bagi sekurang-kurangnya 1 litar bersama dengan simulan aturcara

Untuk menghasilkan simulan litar *input* dan *output*, kita boleh menggunakan laman web Tinkercad secara percuma.

Laman web tinkercad: [www.tinkercad.com](http://www.tinkercad.com)

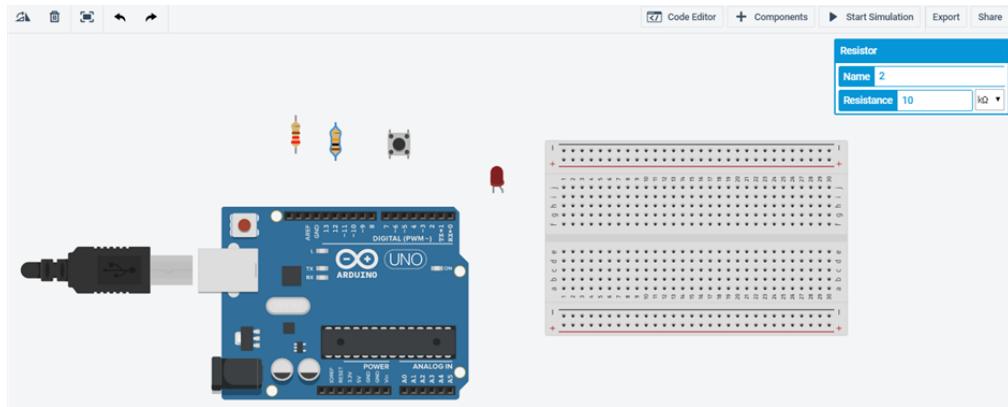
**Litar gabungan input dan output yang melibatkan suis dan LED. Apabila suis ditekan, LED akan menyala.**

- 1) Tambahkan peranti-peranti suis, perintang, LED, Arduino dan juga papan reka dengan butang "+ Components"

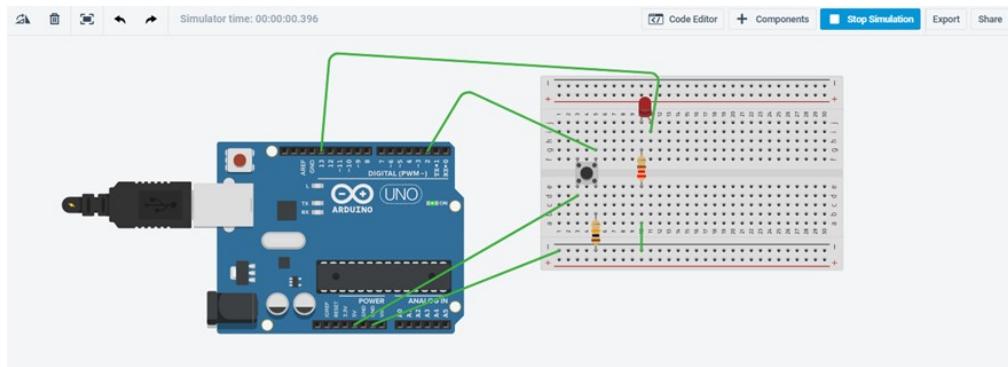


# UNIT 4.3

2) Pastikan nilai perintang adalah betul dengan memilih perintang dan menetapkan nilainya kepada 220 Ohm dan juga 10k Ohm.

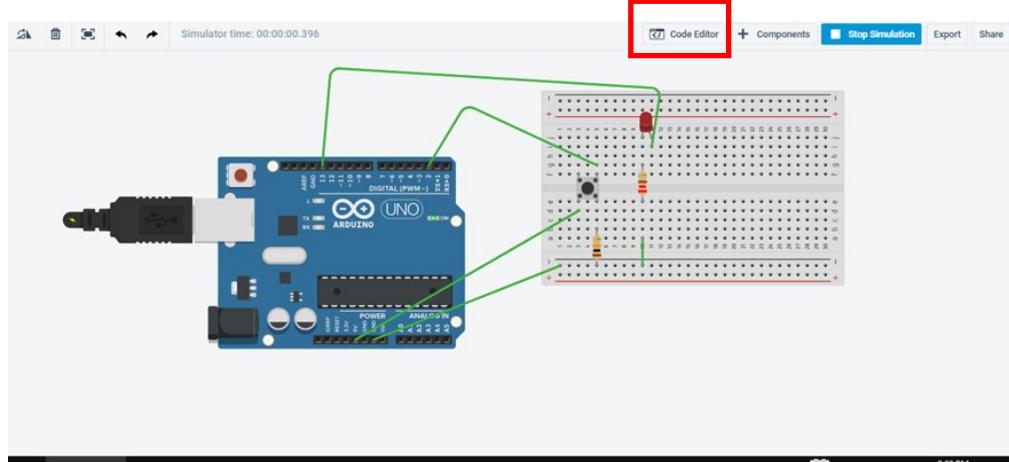


3) Sambungkan peranti-peranti ke atas papan reka dengan kaedah “drag and drop”.



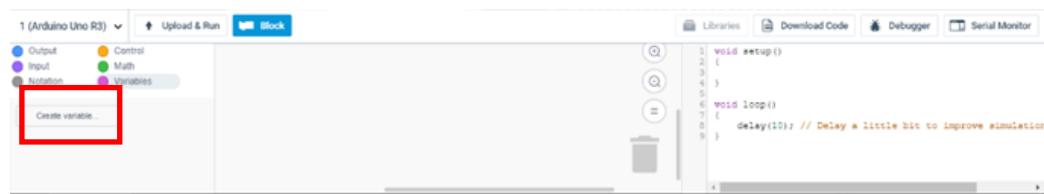
# UNIT 4.3

4) Tuliskan arut cara di bahagian “Code Editor”.



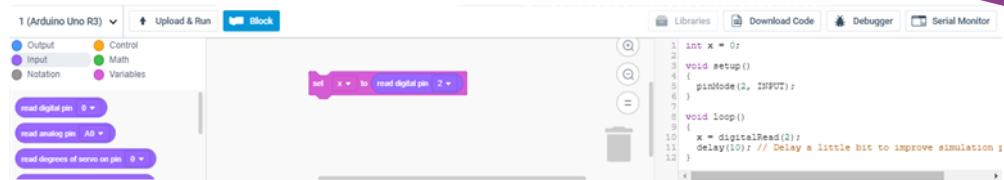
5) "Drag and Drop" blok yang berkenaan untuk hasilkan arut cara bagi membolehkan LED berkelip di pin 5.

a) Hasilkan pemboleh ubah baru

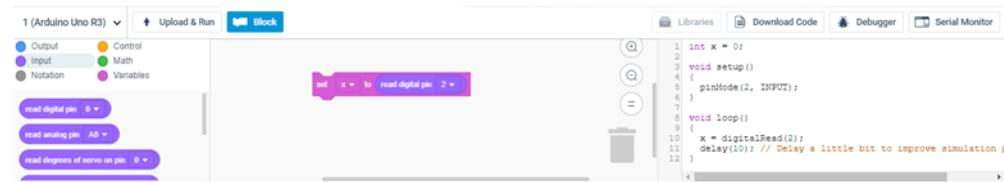


## UNIT 4.3

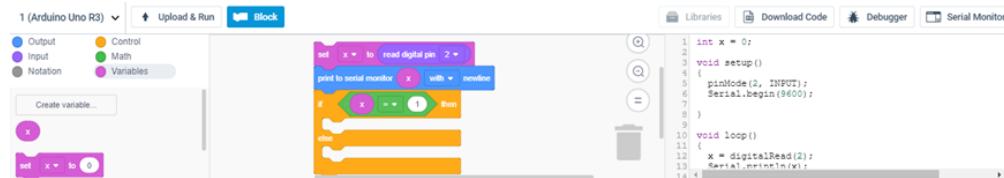
b) Tetapkan nilai pemboleh ubah tersebut kepada nilai bacaan pin 2.



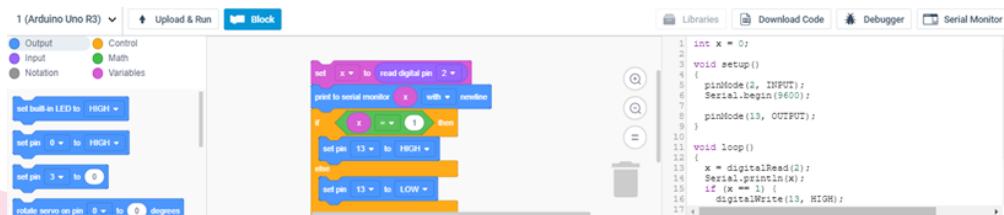
c) Paparkan nilai bacaan dengan komunikasi Serial (gabungan blok *Output* dan blok *Variables*).



d) Hasilkan blok struktur kawalan pilihan dengan memilih blok IF...THEN...ELSE dari blok Control. Kemudian, guna blok Math dan Blok Variable untuk hasilkan gabungan blok berikut:

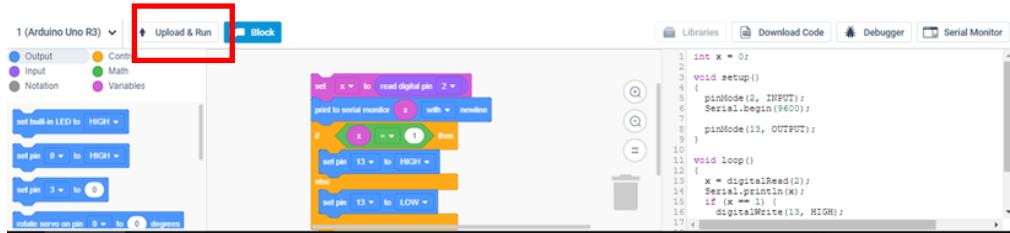


e) Tarik blok untuk nyalakan dan padamkan LED pada pin 13 ke dalam blok struktur kawalan pilihan seperti berikut:

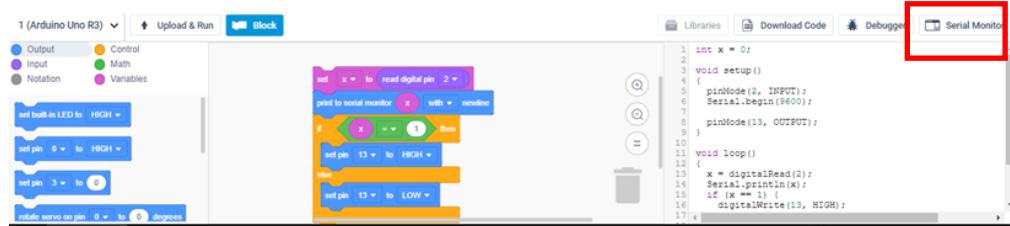


# UNIT 4.3

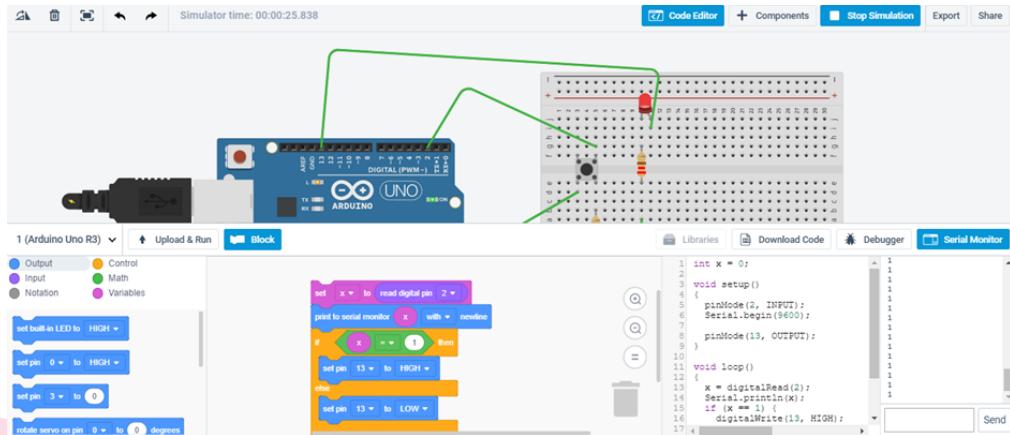
6) Setelah atur cara telah siap disusun, tekan Upload and Run.



7) "Serial Monitor" boleh disimulasikan dengan butang "Serial Monitor".



8) Apabila suis ditekan, LED akan menyala. Nilai pada "Serial Monitor" juga akan berubah.



## UNIT 4.4

# PENGHASILAN PROJEK

### Objektif Pembelajaran

Dalam unit ini, murid akan dapat menghasilkan projek yang melibatkan *input* dan *output*

### Kriteria Kejayaan:

Murid dapat menghasilkan projek yang menghasilkan projek yang melibatkan *input* dan *output* berserta dengan simulasi dan dokumentasi projek

Berikut adalah langkah-langkah untuk menghasilkan projek dengan menggunakan mikropengawal, peranti *input*, peranti *output* dan juga pengaturcaraan.



# Lampiran I

## PENGATURCARAAN

## MENGGUNAKAN TELEFON PINTAR

Langkah-langkah:

1. Tentukan sama ada telefon pintar memiliki fungsi *On-The-Go (OTG)*.
  - Lakukan carian melalui laman sesawang atau membuat carian di ‘Settings’ dalam telefon pintar atau menggunakan aplikasi USB OTG CHECKER in *Google Playstore*<sup>1</sup>.
  - Aktifkan OTG jika perlu.
2. Muat turun aplikasi Arduino Droid<sup>2</sup> di *Google Playstore*.
3. Tulis atur cara yang dikehendaki.
4. Pilih ikon kilat untuk *compile*.
5. Pilih *Settings-> Board Type -> Arduino*  
-> *Uno CH340G (Maker UNO) atau Uno (Arduino)*.
6. Sambungkan mikropengawal kepada telefon pintar dengan menggunakan kabel usb dan kabel OTG.
7. Tekan ikon muat naik untuk menghantar atur cara ke mikropengawal.



<sup>1</sup><https://play.google.com/store/apps/details?id=com.app.usbtogchecker&hl=en>

<sup>2</sup><https://play.google.com/store/apps/details?id=name.antonsmirnov.android.arduinoandroid2&hl=en>

## Lampiran 1

# SENARAI RUJUKAN / SUMBER TAMBAHAN

---

**Untuk maklumat tambahan tentang Arduino:**

- <https://www.arduino.cc/en/Guide/HomePage>
- <https://www.arduino.cc/en/Tutorial/HomePage>
- <http://forefront.io/a/beginners-guide-to-arduino/>
- <https://www.makerspaces.com/arduino-uno-tutorial-beginners/>

**Tutorial video juga akan dimuat naik ke laman web Arus**

- [www.arusacademy.org.my](http://www.arusacademy.org.my)

**Untuk dapatkan idea projek-projek yang boleh dihasilkan:**

- <https://www.makerspaces.com/15-simple-arduino-uno-breadboard-projects/>
- <http://www.instructables.com/id/Arduino-Projects/>

**Lukiskan skematic dengan perisian Fritzing:**

- <http://fritzing.org/home/>