KEMENTERIAN PENDIDIKAN TINGGI
JABATAN PENDIDIKAN POLITEKNIK DAN KOLEJ KOMUNITI

KOLEJ KOMUNITI KUALA PILAH

PENGENALAN

iNTERNET oF tHINGS



LEBAH (SWARM INTELLIGENCE):
Dalam IoT, lebah melambangkan 'Swarm Intelligence' & 'Mesh Networking'—nod sensor berhubung dan bertukar maklumat secara autonomi tanpa kawalan berpusat dan membuat keputusan berdasarkan input

SHARIZAT BIN MOKHTAR

POLYCC®
TVET For All

PENGENALAN

INTERNET OF THINGS



Penulis

SHARIZAT BIN MOKHTAR

Diterbitkan oleh

KOLEJ KOMUNITI
KUALA PILAH

2026

eBuku Pengenalan kepada *Internet Of Things* ini diterbitkan oleh Kolej Komuniti Kuala Pilah.

Diterbitkan oleh :

Kolej Komuniti Kuala Pilah
Pusat Komersial Melang,
72 000, Kuala Pilah,
Negeri Sembilan,
Malaysia.

Tel. : 06-4812545
Fax : 06-4811301

Cetakan Pertama 2026

©Hak cipta terpelihara. Tidak dibenarkan mengeluar ulang mana-mana bahagian teks, ilustrasi dan isi kandungan buku ini dalam apa jua bentuk dan dengan apa jua cara, sama ada secara elektronik, fotokopi, mekanik, rakaman atau cara lain kecuali dengan keizinan bertulis daripada Kolej Komuniti Kuala Pilah, Pusat Komersial Melang,72000, Kuala Pilah, Negeri Sembilan, Malaysia.

All rights reserved. No part of the text, illustration or contents of this book may be reproduced in any form or by any means, whether electronically, photocopying, mechanical, recording or otherwise except with the written permission of Kolej Komuniti Kuala Pilah, Pusat Komersial Melang,72000, Kuala Pilah, Negeri Sembilan, Malaysia.

Perpustakaan Negara Malaysia

Reka bentuk kulit : Muhammad Afiq bin Basiron
Reka bentuk huruf dalaman : Sharizat bin Mokhtar

PENGENALAN INTERNET OF THINGS

eISBN 978-967-18768-9-3

9 789671 876893

KOLEJ KOMUNITI KUALA PILAH

(online)

MANUAL PENGGUNA - AR

EBOOK EXPLORASI PINTAR: PERANTI MUDAH ALIH

E-Book ini mengandungi elemen Augmented Reality (AR) untuk meningkatkan pengalaman pembelajaran secara interaktif. Bagi memastikan anda dapat mengakses kandungan AR dengan lancar, sila ikuti langkah berikut:





✨ 1: MENGIMBAS KOD QR

✅ Buka aplikasi kamera telefon pintar atau aplikasi pengimbas QR:

- Kebanyakan peranti iPhone dan Android menyokong fungsi imbasan QR melalui aplikasi Kamera.
- Halakan kamera ke arah kod QR yang dipaparkan.

✅ Tekan pada pautan atau notifikasi yang muncul bagi membuka paparan imej AR melalui pelayar web (browser) menggunakan platform Assemblr.

✨ 2: MEMBERI KEBENARAN AKSES KAMERA

✅ Gerakkan peranti secara perlahan bagi membantu aplikasi mengesan permukaan rata (meja atau lantai).

✅ Setelah pengesanan berjaya, imej AR akan dipaparkan pada skrin peranti.

✅ Sistem akan meminta kebenaran untuk mengakses kamera peranti anda.

✅ Sila pilih Benarkan bagi membolehkan imej AR dipaparkan dalam persekitaran sebenar melalui kamera.

✨ 3: MEMAPARKAN DAN MELETAKKAN IMEJ AR

✅ Gerakkan peranti secara perlahan bagi membantu aplikasi mengesan permukaan rata (contohnya meja atau lantai).

✅ Setelah pengesanan berjaya, imej AR akan dipaparkan pada skrin peranti.

✨ 4: BERINTERAKSI DENGAN IMEJ AR

✅ Gunakan gerakan pinch pada skrin untuk memvesar atau mengecilkan imej.

✅ swipe untuk memutar imej mengikut keperluan.

✅ Ubah kedudukan peranti untuk melihat imej dari pelbagai sudut perspektif.

Alami Pengalaman AR di halaman berikut :

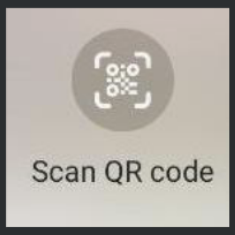
3
5
7
10
20



Paparan AR

ASSEMBLR Studio

01
Imbas QR



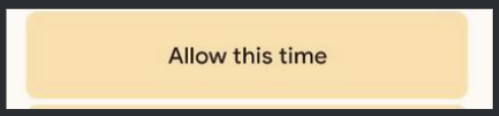
02
Open Link



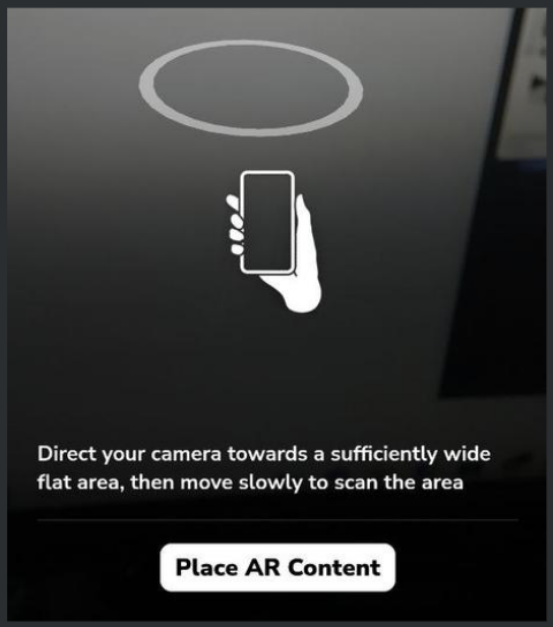
03
Pilih View in 3D



04
Kebenaran Kamera



05
Place AR Content



PRAKATA

Assalamu'alaikum Warahmatullahi Wabarakatuh dan Salam Sejahtera,

Setinggi-tinggi kesyukuran dipanjatkan ke hadrat Allah SWT kerana dengan limpah kurnia-Nya, buku **"Pengenalan kepada Teknologi *Internet of Things* (IoT)"** ini berjaya diterbitkan sebagai bahan bantuan pengajaran dan pembelajaran yang komprehensif. Perkembangan pesat Revolusi Industri 4.0 (IR 4.0) telah menjadikan teknologi *Internet of Things* (IoT) sebagai salah satu kemahiran teras yang wajib dikuasai dalam pendidikan Pendidikan dan Latihan Teknikal dan Vokasional (TVET).

Buku ini ditulis khas untuk dijadikan bahan rujukan utama dan panduan amali berstruktur bagi **pelajar Kolej Komuniti peringkat Sijil**, khususnya dalam bidang Teknologi Maklumat, Teknologi Elektrik, Elektronik, serta bidang kejuruteraan berkaitan. Selain itu, modul ini juga amat sesuai digunakan oleh para pensyarah dan pelajar Kolej Vokasional serta institusi kemahiran tempatan.

Kandungan buku ini disusun secara berperingkat (*step-by-step*) merangkumi tiga bab utama:

i. **Bab 1 (Teori Asas IoT & IR 4.0):** Memahami konsep asas IoT, seni bina SOA dan API, integrasi Kecerdasan Buatan (AIoT), serta sumbangannya terhadap Matlamat Pembangunan Lestari (SDG) UNESCO.
ii. **Bab 2 (Peranti dan Litar IoT):** Menumpukan latihan amali menggunakan mikropengawal **Arduino Uno** yang tahan lasak di bengkel, penyambungan sensor/aktuator, dan ketersambungan rangkaian melalui modul ESP-01.
iii. **Bab 3 (Pembentukan Projek Asas ESP32):** Mengangkat kemahiran pelajar ke peringkat aplikasi awan (*cloud*) melalui amali kawalan LED fizikal pada pin digital D4 secara wayarles menggunakan integrasi platform **Blynk IoT**.

Buku ini direka dengan pendekatan "mesra amali" berasaskan gaya penyampaian yang mudah difahami, dilengkapi dengan rajah litar, ulasan kod program C++, serta panduan penyelesaian masalah (*troubleshooting*) praktikal di bengkel.Terdapat juga bahan berkaitan *Augmented Reality* yang diharap membantu pengajaran.

Penulis ingin merakamkan setinggi-tinggi penghargaan kepada semua pihak yang terlibat secara langsung atau tidak langsung dalam menjayakan penerbitan buku ini. Semoga buku ini mampu menjadi panduan yang bermanfaat, sekali gus melahirkan graduan peringkat Sijil Kolej Komuniti yang berkemahiran tinggi, berdaya saing, dan siap siaga mendepani cabaran industri digital.

Sekian, terima kasih.

Sidang Penulis / Pengarang

ISI KANDUNGAN

ISI KANDUNGAN **7**

BAB 1: PENGENALAN KEPADA INTERNET OF THINGS (IoT) ***9***

1.0 Pengenalan: Apa Sebenarnya IoT? **10**

1.1 Memahami Revolusi Industri: Bagaimana Kita Sampai ke Era IR 4.0? **11**

1.1.1 Sejarah Ringkas Revolusi Industri /*Industrial Revolution* (IR) 11

1.1.2 Sembilan (9) Teras Utama IR 4.0 16

1.1.3 Apakah Kelebihan IR 4.0 kepada Industri? 18

1.2 Memahami Konsep IoT dalam Era IR 4.0 ..... **19**

1.2.1 Konsep Asas IoT: Bagaimana Ia Beroperasi? 19

1.2.2 IoT sebagai "Sistem Saraf" IR 4.0 20

1.2.3 Seni Bina Rangkaian IoT: SOA vs API 20

1.3 Memahami Evolusi dan Perkembangan IoT **22**

1.3.1 Sejarah Perkembangan IoT 22

1.3.2 Sumbangan IoT dalam Dasar SDG UNESCO (Matlamat Pembangunan Lestari) 22

1.3.3 Isu dan Cabaran Utama dalam IoT 23

1.4 Menerangkan Aplikasi IoT dalam Kehidupan **24**

1.4.1 Contoh Aplikasi IoT Nyata 24

1.4.2 Tiga Komponen Utama Teknologi IoT 24

1.5 Membincangkan Fungsi AI dalam IoT (AIoT) **25**

1.5.1 Bentuk AI Berdasarkan Keupayaan Kognitif 26

1.5.2 Jenis AI Berdasarkan Fungsi Kerja 27

1.5.3 Contoh Aplikasi AI dalam IoT (AIoT) 28

1.6 Mentafsir Kajian Kes Berkaitan Aplikasi IoT **28**

1.6.1 Kajian Kes 1: Projek Pengurusan Bencana Banjir Ghana (UNESCO & Kerajaan Jepun) 28

1.6.2 Kajian Kes 2: Pemantauan Kualiti Konkrit Tapak Pembinaan di Malaysia (CIDB & JKR) 29

BAB 2: PERANTI DAN LITAR IoT (FOKUS: ARDUINO UNO) **30**

2.1 Memahami Microcontroller (MCU) **31**

2.1.1 Spesifikasi Perkakasan Arduino Uno sebagai Teras IoT 31

2.1.2 Perisian Microcontroller (MCU) 36

NOTA: Arahan Asas Aturcara Arduino ***36***

a. Struktur Asas Program (*Sketch*) **36**

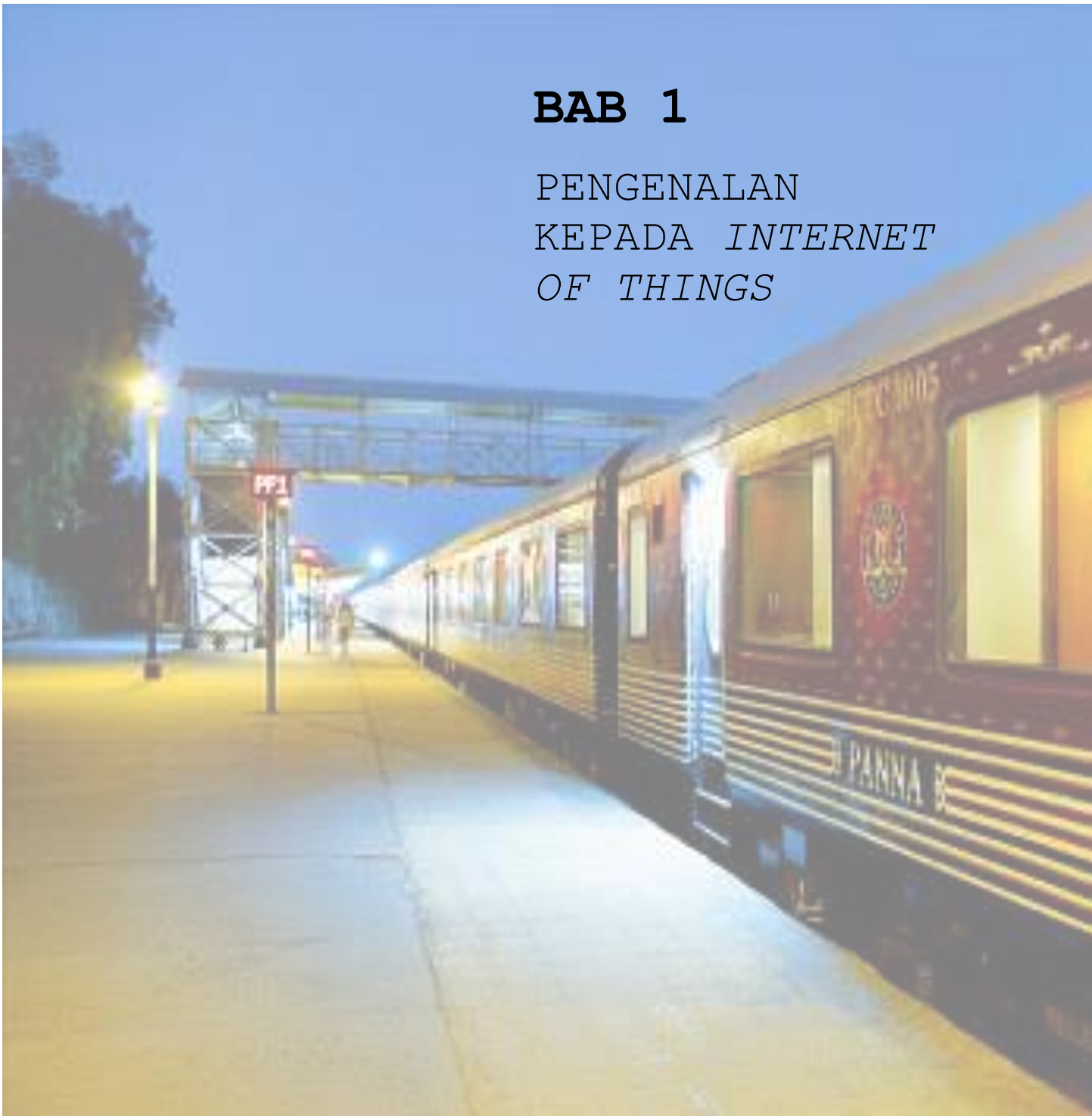
2.2 Menerangkan fungsi *sensor* (penderia) sebagai Peranti IoT **40**

2.2.1 Kepentingan *Sensor* dalam Aplikasi IoT 40

2.2.2 Jenis-jenis Sensor yang Kerap Digunakan di Bengkel 40

2.2.3 Keperluan Kuasa Sensor dan Sambungan ke Arduino I/O 40

2.3 Menerangkan Fungsi Aktuator sebagai Peranti IoT 41
2.3.1 Motor DC dan Relay dalam Aplikasi IoT 41
2.4 Menetapkan Perisian IDE dan Peranti IoT 42
2.4.1 Menyediakan Perisian IDE untuk Arduino Uno 42
2.4.2 Membina Sambungan, Menyusun Semula, dan Memuat Naik Kod 43
2.5 Menetapkan Input dan Output dalam Aplikasi Sebenar dengan Platform IoT 43
2.5.1 Membaca Data dari Peranti Input (Fasa Tempatan) 44
2.5.2 Menghubungkan Arduino Uno ke Platform IoT (*Blynk* Cloud) 45
2.5.3 Paparkan Data ke Peranti Output (Manual & *Blynk Cloud*) 49
2.6 Penyelesaian Masalah (*Troubleshooting*) dan Kesalahan Lazim di Bengkel 50
2.6.1 Isu *Compiler* dan Muat Naik (*Compile & Upload*) 50
2.6.2 Isu Sambungan Rangkaian (*Network & Blynk*) 50
a. Sintaks (*Syntax)(*Cara Tulis Kod Yang Betul) 51
b. Variable (Pembolehubah) 51
c. *Parameter* (Nilai Dalam Fungsi) 52
d. Operator (Simbol Pengiraan & Perbandingan) 52
e. Struktur Kawalan (*Control Structure)* 52
📌 **Ringkasan Cepat *(Quick Recap)* 53**
BAB 3: PEMBENTUKAN PROJEK ASAS DENGAN ESP32 (GABUNGAN BLYNK-IoT) 54
3.1 Pengenalan Ringkas Perkakasan ESP32 56
3.2 Memahami Pin GPIO ESP32 56
3.3 Amali 3A: LED Terbina Dalam (*Built-In* LED Blink) 56
3.4 Amali 3B: Mengawal LED Luaran Menggunakan BlynkIoT (Pin Maya) 65
3.4.1 Perkakasan yang Diperlukan 65
3.4.2 Rajah Sambungan Litar 65
3.4.3 Kod Program Amali 3B: BlynkBlink LED Luaran (ESP32 WiFi) 66
3.4.4 Panduan Penyediaan *Widget* Suis pada Aplikasi *Blynk* 66



BAB 1

PENGENALAN KEPADA *INTERNET OF THINGS*

This Photo by Unknown Author is licensed under CC BY

BAB 1: PENGENALAN KEPADA *INTERNET OF THINGS* (IoT)

1.0 Pengenalan: Apa Sebenarnya IoT?

Bayangkan anda sedang dalam perjalanan pulang ke rumah pada hari yang sangat panas. Sebelum sampai ke rumah, anda hanya perlu membuka telefon pintar dan menghidupkan peralatan elektrik seperti kipas di bilik tidur. Apabila anda melangkah masuk ke rumah, bilik sudah pun sejuk dan selesa. Di dapur pula, peti sejuk anda boleh mengesan jika susu kotak sudah habis dan secara automatik menghantar senarai barangan runcit ke telefon anda.

Inilah yang kita panggil sebagai ***Internet of Things*** **(IoT)** atau **Internet Benda**. Secara mudahnya, IoT ialah satu konsep di mana kita mengambil "benda" atau objek fizikal di sekeliling kita sama ada lampu, kipas, pendingin hawa , pasu bunga, kereta atau almari seterusnya pasangkan penderia (*sensor*) dan serta sambungan internet kepada objek tersebut. Dengan cara ini, objek-objek tersebut boleh "bercakap" dengan menghantar data ke awan (*cloud*), berinteraksi sesama sendiri, dan menjalankan arahan secara automatik tanpa kita perlu menekan suis secara fizikal.

Nota: Pautan video pengenalan

https://www.youtube.com/watch?v=-icFb4rgnPA&t=6s

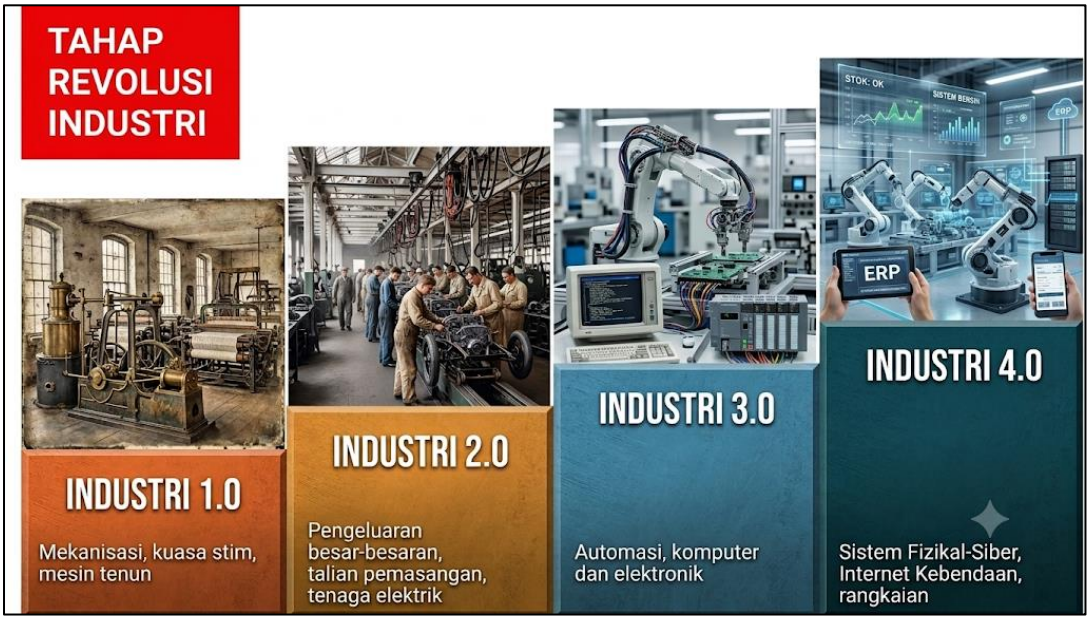
1.1 Memahami Revolusi Industri: Bagaimana Kita Sampai ke Era IR 4.0?

1.1.1 Sejarah Ringkas Revolusi Industri /*Industrial Revolution* (IR)

Sebelum kita menyelami IoT dengan lebih dalam, mari kita tengok dahulu bagaimana sejarah industri manusia berubah. Kita tidak sampai ke era yang hebat ini dalam sekelip mata. Manusia telah melalui empat fasa perubahan besar yang kita panggil sebagai Revolusi Industri yang bermula di Eropah yang menyebabkan pembukaan bandar-bandar besar sebagai pusat industri dan penghijrahan penduduk untuk mencari pekerjaan :

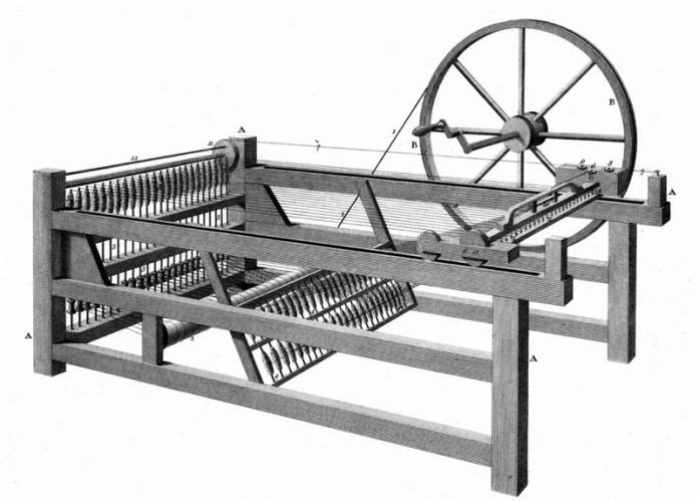
Jadual 1.1. Era perubahan dalam Revolusi Industri

Masa	Peristiwa
Akhir Abad ke-18	IR1.0 Penggunaan Kuasa Wap & Automasi Manual dengan komponen mekanikal Mekanisasi Jentera (Mekanikal)
Awal Abad ke-20	Tenaga Elektrik & Pengeluaran Besar-besaran (Elektrikal)
Pertengahan Abad ke-20	Komputer, Automasi Awal & Teknologi Maklumat (IR 3.0) (Komputer & Mekatronik)
Abad ke 21	Sistem Siber-Fizikal, IoT, Awan & Kecerdasan Buatan (IR 4.0) (digitalisasi)

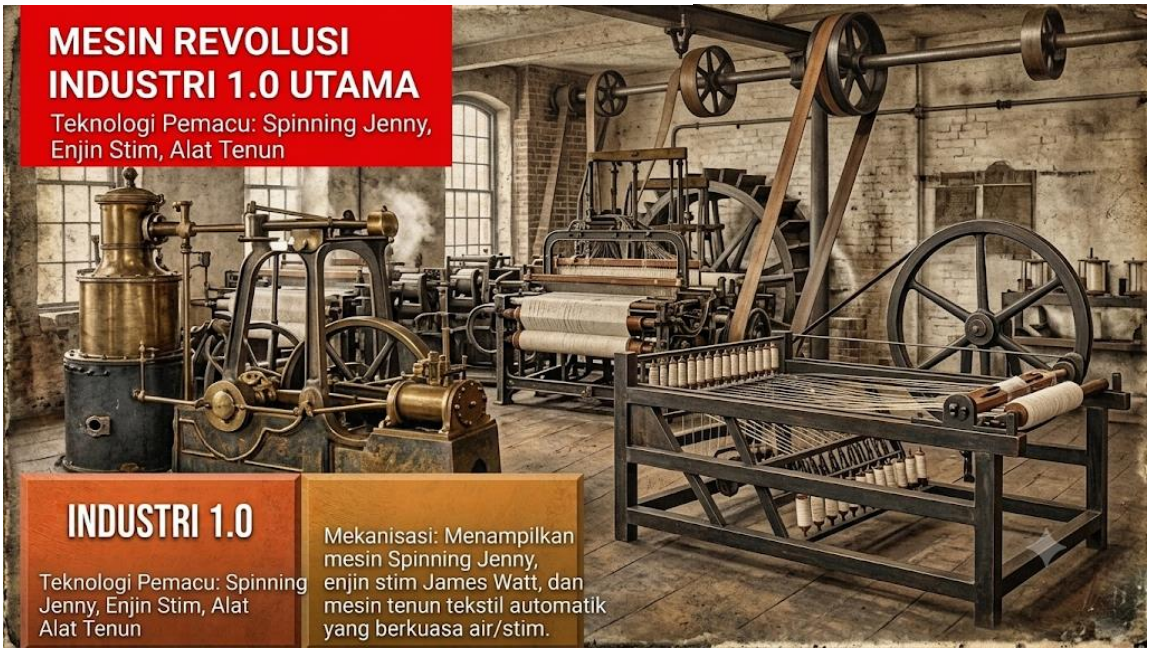


Rajah 1.1 : Tahap revolusi industri dalam sejarah dunia.

a. **Revolusi Industri Pertama (IR 1.0):** Bermula pada akhir abad ke-18 di Britain. Pada waktu ini, manusia mula beralih daripada menggunakan tenaga otot (manusia dan haiwan) kepada jentera mekanikal yang digerakkan oleh **kuasa wap dan air**. Penemuan ini mengubah cara kain ditenun dan barang dihasilkan di kilang-kilang awal.



Rajah 1.2 : *Spinning Jenny* yang memutarkan sehingga 8 gelung benang berbanding hanya 1 gelung melalui kaedah sebelumnya :https://www.britannica.com/technology/spinning-jenny



Rajah 1.3 : Mesin-mesin dalam revolusi perindustrian pertama

b. **Revolusi Industri Kedua (IR 2.0):** Bermula pada awal abad ke-20. Era ini dimulakan dengan penemuan seterusnya penggunaan **tenaga elektrik**. Dengan adanya tenaga baharu ini, kilang-kilang mula menggunakan mesin berkuasa elektrik dan konsep laluan pembuatan atau *assembly line* diperkenalkan. Melalui kaedah ini , produk dapat dihasilkan dalam jumlah yang sangat besar dan cepat pada kos yang lebih murah. Kereta Ford Model T adalah contoh klasik produk era ini.



Rajah 1.3 : Penemuan elektrik menjadi satu lambang kemajuan teknologi IR2.0.



Rajah 1.4 : *Assembly Line di* kilang Ford pada awal pembinaan . Wikimedia Commons/public domain



Scan QR ini menggunakan kamera anda untuk askes ke objek menggunakan *Augemented Reality*

https://asblr.com/cqkF8Cf

c. **Revolusi Industri Ketiga (IR 3.0):** Bermula sekitar tahun 1960-an. Ini merupakan era lahirnya **komputer, elektronik, dan internet**. Komputer menjadi sumber utama untuk sesebuah operasi dan bukan sekadar membantu tugasan harian.

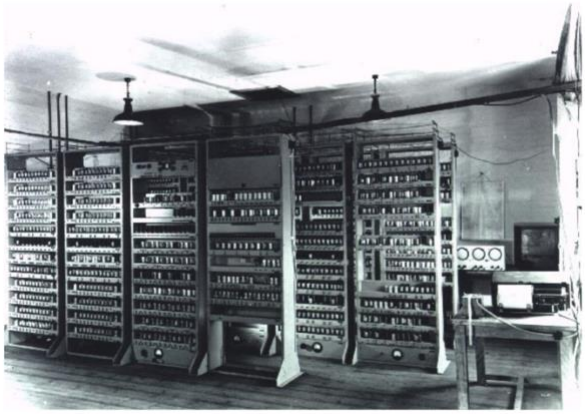


Rajah 1.5 : Komputer yang menjadi pemacu utama kemajuan dalam IR3.0.

Automasi mula diperkenalkan di mana mesin di kilang tidak lagi dioperasikan sepenuhnya oleh manusia, sebaliknya dikawal oleh pengawal logik boleh atur cara (*Programmable Logic Controller* - PLC) dan komponen mekatronik.



Rajah 1.6 : Pengawal Logik boleh aturcara (PLC) merupakan pengawal yang boleh diprogramkan untuk mesin beroperasi mengikut arahan. *Omron Europe*



Rajah 1.7: EDSAC merupakan antara computer terawal dibangunkan di Universiti of Cambridge . https://www.cst.cam.ac.uk/news/70-years-first-computer-designed-practical-everyday-use



Rajah 1.8 : Automasi menggunakan komponen mekatronik di kilang pembuatan .

https://activeyouth.lt/2024/01/31/understanding-industry-3-0/?lang=en

d. **Revolusi Industri Keempat (IR 4.0):** Inilah era yang sedang kita lalui sekarang. Perbezaan utama IR 4.0 dengan era sebelum ini ialah sambungan semua peralatan dan operasi melalui internet. Data menjadi komoditi terpenting untuk zaman ini. Kita tidak lagi hanya menggunakan komputer untuk bekerja, tetapi menghubungkan dunia fizikal (mesin, *sensor*, pemproses) dengan dunia siber (internet, awan, AI). Mesin kini boleh menghantar data , berkomunikasi sesama sendiri dan membuat analisa dan membuat keputusan tanpa menunggu arahan manusia untuk tugasan ringkas.



Rajah 1.9 : Robot Alice sebagai robot pemberi maklumat di lapangan terbang

Industri 4.0 ini juga adalah tentang penemuan pelbagai teknologi baharu yang antara lain menggunakan automasi untuk analisis dan data raya (*big data)*, simulasi, integrasi sistem, penggunaan robotik secara meluas sehingga dalam kelengkapan domestik , *cloud*, *Internet of Things (IoT),* dan perkara yang seumpamanya.

1.1.2 Sembilan (9) Teras Utama IR 4.0

Untuk membina ekosistem industri yang benar-benar pintar, terdapat sembilan teknologi sokongan yang menjadi tiang atau tonggak utama IR 4.0:

i. **Robotik Auto (*Autonomous Robots*):** Robot dalam IR 4.0 bukan lagi sekadar tangan mekanikal yang kaku. Ia dilengkapi dengan AI untuk bekerjasama secara selamat di sebelah manusia (*cobots*), mudah menyesuaikan diri dengan tugasan baharu, dan boleh membuat keputusan operasi secara mandiri.

ii. **Simulasi (*Digital Twin*):** Sebelum kita membina bangunan, jambatan, atau enjin jet yang mahal, kita bina dahulu "kembar digital" atau replika maya di dalam komputer. Sebarang ujian, tekanan, dan simulasi dilakukan pada model maya ini untuk meramal kerosakan dan mengoptimumkan prestasi tanpa merosakkan perkakasan fizikal.

iii. **Sistem berintegrasi (*Horizontal and Vertical Integration*):** Di dalam kilang pintar, tiada jabatan yang bekerja secara sendirian. Data daripada pembekal bahan mentah, rantaian logistik, mesin pembuatan di kilang, sehinggalah kepada maklum balas pelanggan akhir dihubungkan dalam satu rangkaian data yang lancar.

iv. **Perindustrian berteraskan Internet of Things (IoT):** Pemasangan penderia pintar pada setiap mesin dan peralatan kilang untuk mengumpul data prestasi, suhu, dan getaran secara berterusan.

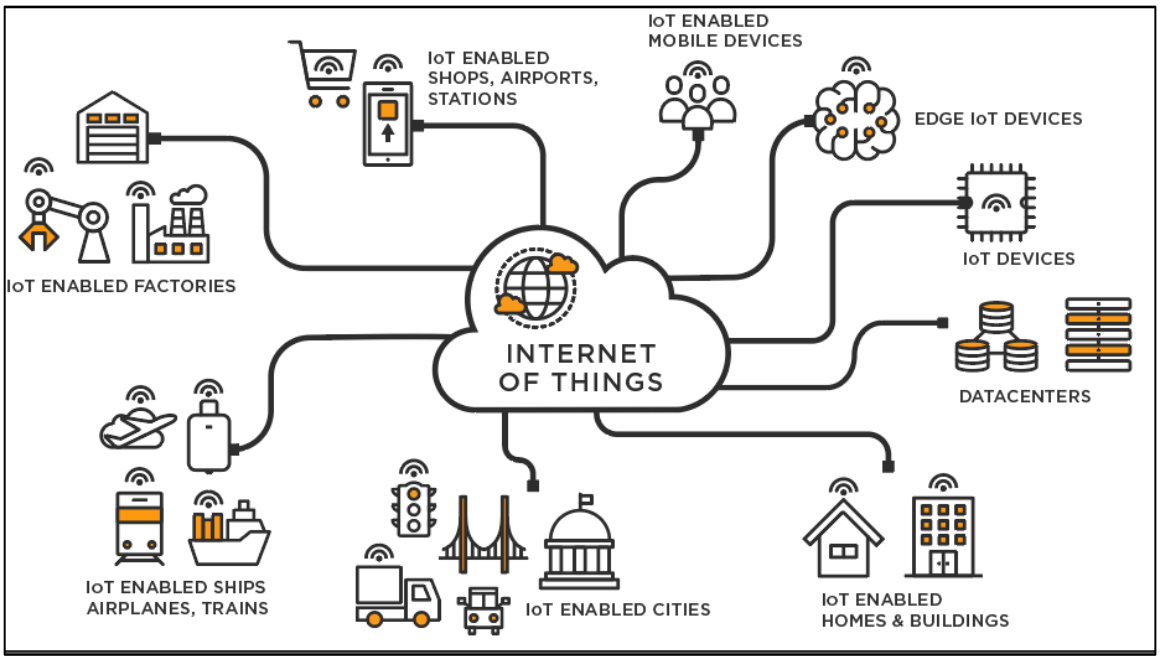
v. **Keselamatan Siber (*Cybersecurity*):** Apabila semua peranti dihubungkan ke internet, pintu ancaman penggodaman terbuka luas. Oleh itu, sistem pertahanan siber yang canggih sangat kritikal untuk melindungi data industri dan mengelakkan sabotaj operasi.

vi. **Pengkomputeran Awan (*Cloud Computing*):** Tempat menyimpan dan memproses data berskala besar. Awan membolehkan data dari seluruh dunia diakses dengan pantas, selamat, dan pada bila-bila masa tanpa perlu membina pelayan (*server*) fizikal yang mahal di pejabat sendiri.

vii. **Teknik Pembuatan Tambahan (*Additive Manufacturing / 3D Printing*):** Teknologi mencetak barangan terus daripada fail digital 3D menggunakan plastik, polimer, atau logam. Ini sangat memudahkan penghasilan prototaip pantas dan alat ganti tersuai.

viii. **Realiti Maya Diperluas (*Augmented Reality – AR*):** Teknologi yang memaparkan maklumat atau visual grafik komputer di atas dunia nyata. Contohnya, juruteknik boleh memakai cermin mata AR untuk melihat panduan langkah demi langkah menyelenggara enjin yang terpapar terus di atas enjin fizikal tersebut.

ix. **Analisis Data Raya (*Big Data Analytics*):** Menapis dan menganalisis jutaan gigabait data yang dikumpul oleh sensor IoT untuk mencari corak tersembunyi, trend pasaran, atau tanda-tanda awal kerosakan mesin.



Rajah 1.10 : Gambaran asas *Internet of Things*

1.1.3 Apakah Kelebihan IR 4.0 kepada Industri?

Kenapa industri sanggup berhabis jutaan ringgit untuk beralih kepada IR 4.0? Jawapannya mudah: **kelangsungan dan kecekapan**.

i. **Kecekapan Maksimum:** Penggunaan sensor membolehkan kita mengesan pembaziran tenaga atau bahan mentah secara serta-merta dan terus membuat perubahan dalam perancangan akan datang.

ii. **Penyelenggaraan Ramalan (*Predictive Maintenance*):** Kita tidak lagi menunggu mesin rosak baru hendak baiki (penyelenggaraan reaktif). Sensor akan mengesan getaran luar biasa atau kenaikan suhu pada motor elektrik, memberi amaran awal supaya penyelenggaraan dibuat sebelum mesin benar-benar rosak. Ini menjimatkan kos dan mengelakkan kilang terhenti operasi.

iii. **Kebolehan Pengubahsuaian**: Pengeluar boleh mengubah suai spesifikasi produk di laluan pemasangan dengan cepat mengikut tempahan pelanggan secara individu tanpa perlu menutup keseluruhan sistem operasi kilang.

1.2 Memahami Konsep IoT dalam Era IR 4.0

1.2.1 Konsep Asas IoT: Bagaimana Ia Beroperasi?

Untuk memahami IoT dengan mudah, bayangkan ia berfungsi seperti tubuh manusia:

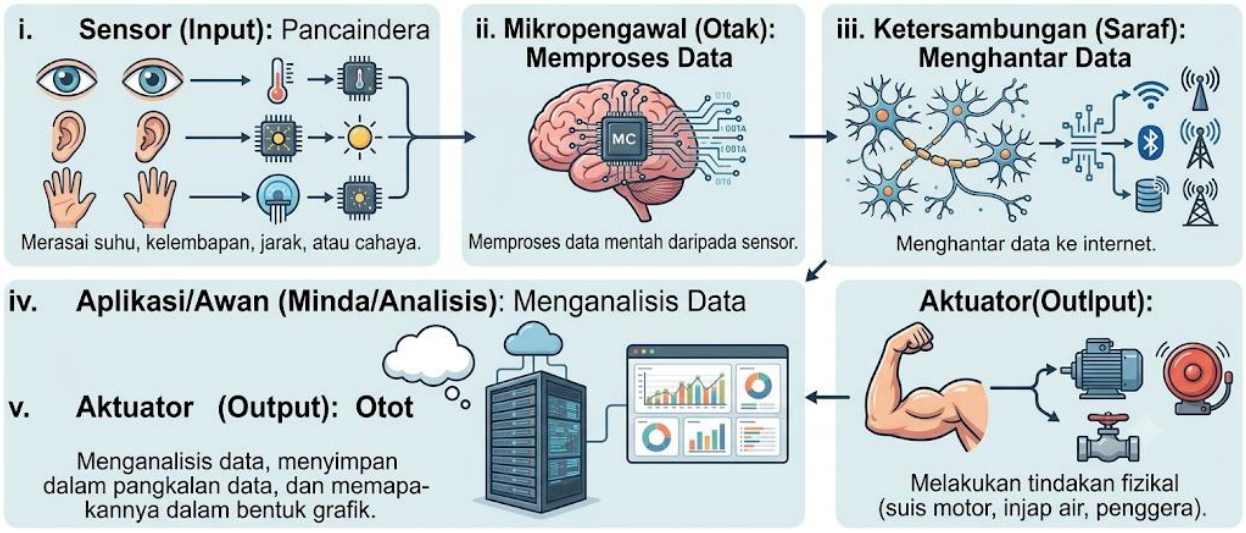
i. **Sensor (Input):** Bertindak seperti pancaidera kita (mata, telinga, kulit) yang merasai suhu, kelembapan, jarak, atau cahaya di sekeliling.

ii. **Mikropengawal (Otak):** Memproses data mentah daripada sensor.

iii. **Ketersambungan (Saraf):** Menghantar data tersebut ke internet melalui Wi-Fi, Bluetooth, atau rangkaian selular.

iv. **Aplikasi/Awan (Minda/Analisis):** Menganalisis data, menyimpan dalam pangkalan data, dan memaparkannya dalam bentuk grafik yang mudah difahami oleh pengguna.

v. **Aktuator (Output):** Bertindak seperti otot kita yang melakukan tindakan fizikal (menghidupkan suis motor, menutup injap air, atau membunyikan penggera).

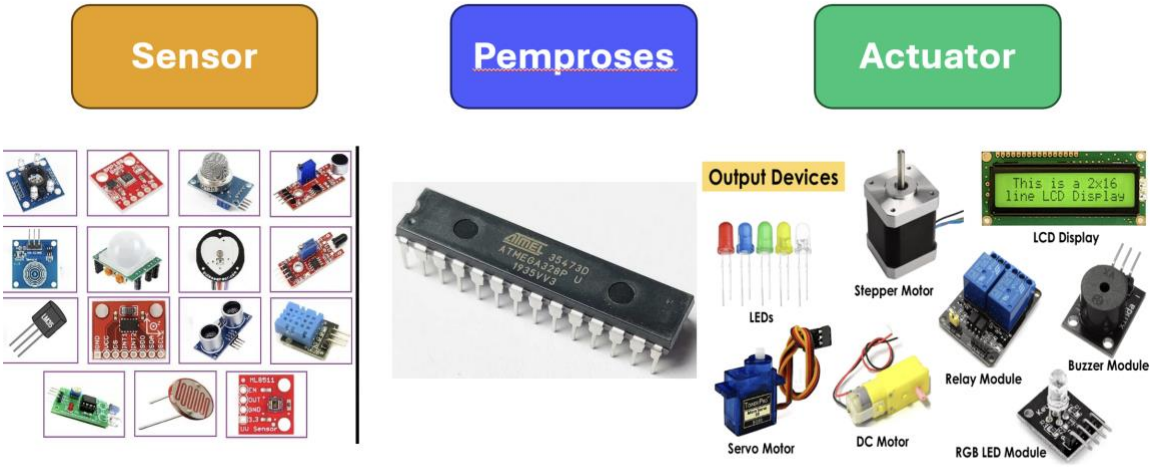


Rajah 1.11 Gambaran analogi IoT berdasarkan tubuh manusia

NOTA: lihat video interaktif di sini https://youtu.be/X6sxksb2JQ8

1.2.2 IoT sebagai "Sistem Saraf" IR 4.0

Dalam tubuh IR 4.0, IoT is rangkaian sarafnya. Tanpa IoT, teknologi seperti Kecerdasan Buatan (AI) atau Analitik Data Raya akan menjadi "lumpuh" kerana tiada bekalan data masa nyata untuk diproses. IoT memastikan setiap mesin, sensor, dan peranti menghantar aliran data berterusan ke pusat pemprosesan untuk dianalisis dan diambil tindakan yang sepatutnya untuk kemudahan pengguna dan kepentingan pembuatan.



Rajah 1.12 : Beberapa komponen yang jatuh di bawah kategori yang dinyatakan

1.2.3 Seni Bina Rangkaian IoT: SOA vs API

Bagaimana peranti IoT berkomunikasi dan diintegrasikan dengan sistem pengisian perisian yang besar? Terdapat dua pendekatan seni bina utama yang biasa digunakan:

i. Seni Bina Berorientasikan Perkhidmatan (*SOA - Service-Oriented Architecture*)

Seni bina SOA ialah pendekatan di mana setiap komponen atau peranti IoT dianggap sebagai satu **"perkhidmatan"** (*service*) yang berdiri sendiri. Setiap perkhidmatan ini mempunyai fungsi yang spesifik (contoh: "Perkhidmatan Ambil Bacaan Suhu", "Perkhidmatan Hidupkan Suis Lampu").

- **Gandingan Longgar (*Loose Coupling*):** Setiap perkhidmatan tidak bergantung rapat antara satu sama lain. Jika satu perkhidmatan mengalami ralat atau dinaik taraf, ia tidak akan meruntuhkan keseluruhan sistem.

- **Abstraksi Tinggi (*Abstraction*):** Pengguna perkhidmatan tidak perlu tahu jenis sensor apa yang digunakan atau bahasa pengaturcaraan apa yang ditulis di dalam peranti; mereka hanya perlu tahu cara memanggil perkhidmatan tersebut.
- **Kepentingan dalam IoT:** SOA sangat berguna dalam ekosistem IoT yang mempunyai peranti daripada pelbagai jenama dan protokol yang berbeza (*heterogen*). Dengan menyalutkan fungsi peranti-peranti ini sebagai perkhidmatan yang standard di lapisan perantaraan (*middleware*), integrasi sistem menjadi jauh lebih mudah.

ii. Seni Bina Berasaskan API (*Application Programming Interface*)

Seni bina berasaskan API adalah kaedah yang lebih moden, ringan, dan popular digunakan pada hari ini untuk menghubungkan peranti IoT terus ke aplikasi web atau awan. API bertindak sebagai jambatan atau pintu masuk rasmi yang membolehkan perisian luaran bertukar-tukur data dengan peranti IoT kita.

- ***RESTful API & JSON:*** Standard industri hari ini banyak menggunakan protokol *HTTP/REST* dengan format data **JSON** (*JavaScript Object Notation*). Format JSON ini sangat ringan, berasaskan teks, dan mudah dibaca oleh manusia serta diproses oleh komputer.
- **Perbezaan Mudah SOA vs API:**
    - **SOA** ialah satu konsep reka bentuk sistem secara makro (skala besar) yang menentukan bagaimana perkhidmatan ditadbir urus dan diuruskan di peringkat perusahaan.
    - **API** pula ialah alat atau titik sambung (*endpoint*) fizikal yang digunakan untuk melaksanakan komunikasi tersebut secara praktikal. Dalam erti kata lain, API boleh digunakan untuk membina dan menghubungkan perkhidmatan di dalam seni bina SOA.

1.3 Memahami Evolusi dan Perkembangan IoT

1.3.1 Sejarah Perkembangan IoT

Idea menghubungkan peranti fizikal ke internet sebenarnya telah bermula sejak awal tahun 1980-an, apabila sekumpulan pelajar di *Carnegie Mellon University* memasang penderia mikro pada mesin pencuci tin air soda *Coca-Cola* untuk memantau sama ada minuman di dalamnya sejuk atau kosong sebelum mereka berjalan ke mesin tersebut.

Namun, istilah ***"Internet of Things"*** hanya dilahirkan secara rasmi pada tahun **1999** oleh **Kevin Ashton** semasa bekerja di *Procter & Gamble* (P&G).



Rajah 1.13 : Pengasas konsep IoT, Kevin Ashton

Beliau menggunakan istilah ini untuk menerangkan bagaimana teknologi RFID (*Radio-Frequency Identification*) boleh digunakan untuk menjejaki rantaian bekalan barangan melalui internet.[1] Sejak itu, dengan pengenalan alamat internet IPv6 (yang menawarkan bilangan alamat IP yang hampir tanpa had) serta penurunan kos sensor dan mikropengawal, IoT telah berkembang daripada projek eksperimen makmal menjadi sebahagian daripada kehidupan harian kita.

1.3.2 Sumbangan IoT dalam Dasar SDG UNESCO (Matlamat Pembangunan Lestari)

Teknologi IoT memainkan peranan yang sangat penting dalam menyokong Matlamat Pembangunan Lestari (*Sustainable Development Goal* - SDG) yang digariskan oleh PBB dan UNESCO untuk membina masa depan bumi yang mampan dan adil di dalam elemen-elemen yang digariskan di bawah.

i. **SDG 4 (Pendidikan Berkualiti):** IoT membolehkan pelajar, khususnya dalam bidang TVET, mendapat akses kepada alat pembelajaran praktikal berkos rendah seperti **Arduino Uno**. Ini melatih kemahiran pemikiran perkomputeran (*computational thinking*) dan penyelesaian masalah realistik di dalam komuniti.

ii. **SDG 6 (Air Bersih dan Sanitasi):** Penggunaan penderia IoT untuk memantau tahap kualiti air, mengesan sisa toksik di sungai, dan menguruskan pengagihan air bersih secara cekap bagi mengelakkan pembaziran sumber utiliti.

iii. **SDG 11 (Bandar dan Komuniti Mampan):** Melalui konsep bandar pintar (*smart city*). Contohnya, lampu jalan pintar yang dimalapkan secara automatik apabila tiada lalu lintas, dan tong sampah pintar yang menghantar isyarat kepada lori sampah hanya apabila tong tersebut sudah 90% penuh, mengurangkan jejak karbon kenderaan pengurusan sisa.

iv. **SDG 13 (Tindakan Terhadap Iklim):** Stesen penceritaan cuaca mikro berasaskan IoT dipasang di kawasan berisiko untuk memantau kelembapan tanah, perubahan suhu ekstrem, dan paras air sungai bagi memberikan amaran awal bencana banjir atau kemarau kepada komuniti setempat.

1.3.3 Isu dan Cabaran Utama dalam IoT

Walaupun IoT menawarkan potensi yang luar biasa, pembangun teknologi hari ini berhadapan dengan beberapa cabaran besar:

i. **Keselamatan Siber dan Privasi:** Banyak peranti IoT dipasarkan tanpa ciri keselamatan yang kukuh.[2] Penggodam boleh mengeksploitasi kelemahan ini untuk menceroboh rangkaian rumah, mencuri data peribadi, atau melakukan serangan penolakan perkhidmatan secara besar-besaran (DDoS).

ii. **Isu pengoperasian (Interoperability):** Ketiadaan satu piawaian industri yang tunggal menyebabkan peranti daripada jenama A selalunya tidak boleh bercakap dengan peranti daripada jenama B. Ini memaksa pembangun menulis kod yang perlu memahami antara satu protokol dengan protokol yang lain menggunakan API yang berlainan.

iii. **Pengurusan Data :** Berbilion peranti IoT menghantar data setiap saat. Mengurus, menapis, dan menganalisis aliran data yang sangat besar ini memerlukan infrastruktur storan dan kuasa pemprosesan yang sangat tinggi pada perkakasan yang menerima data.

iv. **Kekurangan Pakar Tempatan:** Industri kini menghadapi jurang besar dalam mencari jurutera yang mahir dalam gabungan tiga bidang utama: perkakasan elektronik, pengaturcaraan perisian, dan rangkaian keselamatan siber. Bidang kejuruteraan baharu perlu diwujudkan untuk mengisi kekosongan ini.

KUIZ 1 : https://sites.google.com/kkkp.edu.my/ebook-internetofthings/home

1.4 Menerangkan Aplikasi IoT dalam Kehidupan

1.4.1 Contoh Aplikasi IoT Nyata

i. **Rumah Pintar (*Smart Home*):** Suis pintar, termostat automatik, dan sistem keselamatan kamera pintar yang menghantar klip video terus ke aplikasi *Telegram* pemilik apabila mengesan pencerobohan.

ii. **Pertanian Pintar (*Smart Agriculture*):** Sensor analog untuk membaca kelembapan tanah dan pemproses yang mengawal injap solenoid untuk tangki air. Air hanya disiram apabila tanah dikesan benar-benar kering, menjimatkan air dan mengelakkan akar pokok reput akibat penyiraman berlebihan.

iii. **Kesihatan Pintar (*Smart Healthcare*):** Jam tangan pintar yang memantau degupan jantung pesakit jantung secara berterusan. Sekiranya degupan jantung jatuh ke tahap kritikal, ia secara automatik menghantar amaran kecemasan bersama koordinat GPS pesakit kepada doktor dan ahli keluarga. Contoh sedia ada seperi *Apple Watch.*

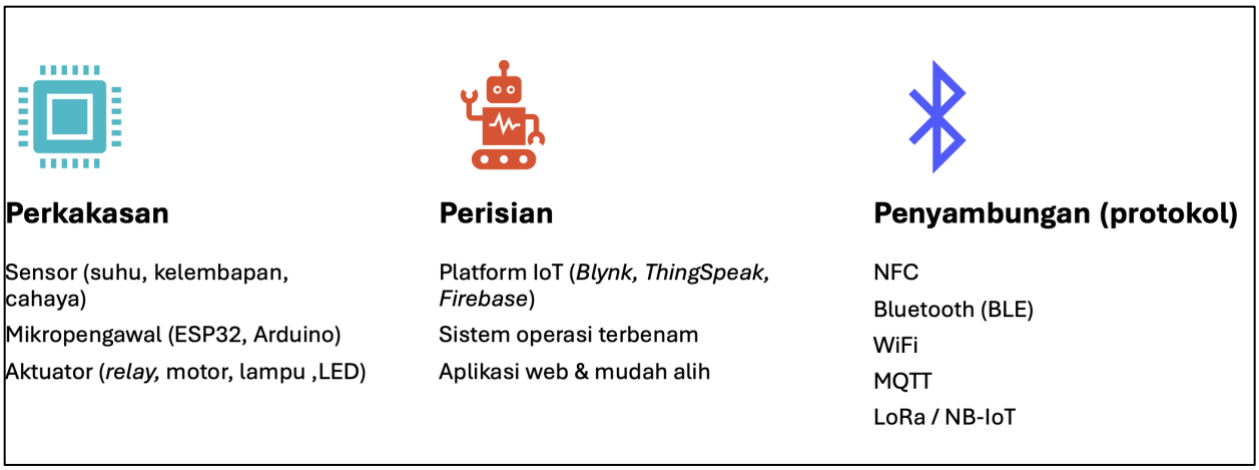
iv. **Logistik Pintar:** Memantau suhu lori kontena yang membawa vaksin atau makanan beku untuk memastikan rantaian sejuk tidak terputus sepanjang perjalanan.

1.4.2 Tiga Komponen Utama Teknologi IoT

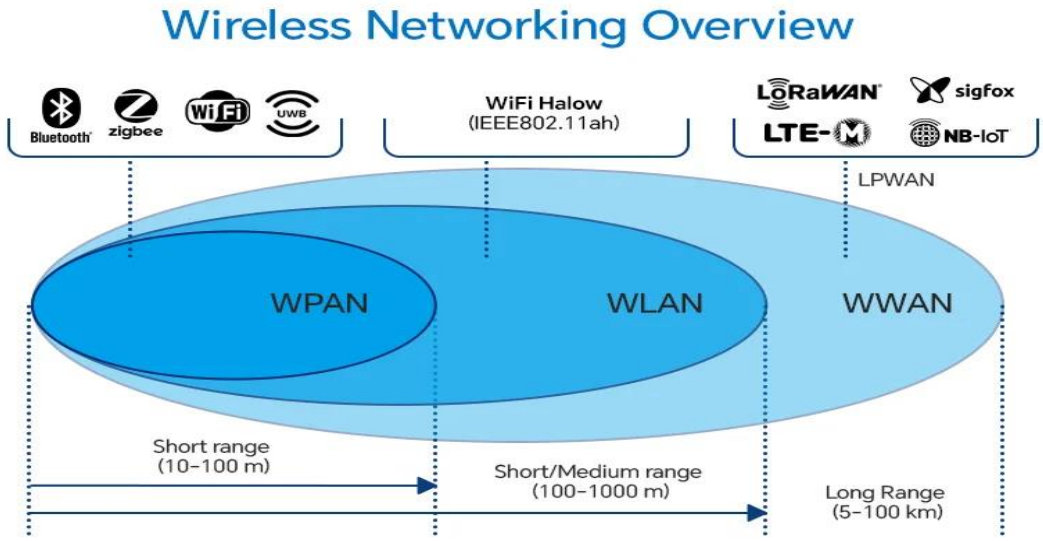
i. **Perkakasan (*Hardware*):** Segala komponen fizikal yang boleh disentuh. Termasuklah penderia (DHT11, LDR), mikropengawal (*Arduino Uno*, ESP32, *Raspberry Pi Pico*), dan aktuator (geganti, motor servo, *buzzer*).

ii. **Perisian (*Software*):** Jiwa bagi perkakasan. Ini melibatkan perisian tegar (*firmware*) yang kita kodkan ke dalam mikropengawal menggunakan Arduino IDE, platform awan untuk menerima data (cth., *Blynk, ThingsBoard*), dan pangkalan data siri masa (*InfluxDB*).

iii. **c. Penyambungan (*Connectivity*):** Rangkaian yang menghubungkan perkakasan dengan perisian. Termasuklah media fizikal/wayarles (Wi-Fi, *Bluetooth Low Energy*, LoRaWAN) dan protokol komunikasi data (MQTT, HTTP).



Rajah 1.14 : komponen dalam sambungan IoT yang jatuh abwah kategori yang berkaitan



Rajah 1.15 : Jenis sambungan bagi jarak dekat dan jauh untuk IoT. Sila maklum bahawa *Narrowband* IoT (NB-IoT) dan *Low Power WAN* (LpWAN) turut digunakan bagi sambungan jarak jauh.

https://www.mokosmart.com/iot-networking-technologies-guide/

1.5 Membincangkan Fungsi AI dalam IoT (AIoT)

Apabila kita menggabungkan **Kecerdasan Buatan (AI)** dengan **Internet of Things (IoT)**, kita melahirkan teknologi baharu yang dipanggil **AIoT (*Artificial Intelligence of Things*)**. Jika IoT bertindak sebagai sistem saraf yang mengumpul data, AI bertindak sebagai otak yang memberikan keupayaan berfikir, meramal, dan membuat keputusan pintar secara autonomi kepada peranti tersebut. Hasilnya ialah sebuah sistem yang dapat membuat keputusan yang tepat dan memudahkan manusia dalam urusan seharian.

1.5.1 Bentuk AI Berdasarkan Keupayaan Kognitif

Berdasarkan sejauh mana keupayaan inteleknya, AI boleh dibahagikan kepada tiga bentuk:

1. **AI Terhad (*Narrow AI* / *Weak AI*):** Ini adalah jenis AI yang kita selalu guna sekarang. AI jenis ini sangat bagus untuk buat kerja spesifik, tapi tak boleh buat benda lain di luar skop latihannya. Contohnya: *ChatGPT, Siri* dan *Alexa* dan sistem pengenalan wajah pada telefon pintar.
2. **AI Am (*General AI* / *Strong AI* / *AGI*):** Tahap AI teori di mana mesin mempunyai keupayaan kognitif, kesedaran diri, dan pemikiran yang setaraf dengan manusia. Ia mampu belajar, merancang, dan menyelesaikan pelbagai masalah yang berbeza secara bebas tanpa perlu diprogramkan semula. Setakat ini, General AI masih belum wujud sepenuhnya dan masih dalam peringkat kajian.
3. **Super AI (*Artificial Superintelligence – AS*):** Tahap spekulatif AI di mana keupayaan intelektual dan pemikiran komputer jauh melampaui keupayaan gabungan otak seluruh manusia yang paling pintar di bumi. Super AI digambarkan mempunyai kesedaran diri penuh, emosi sendiri, dan keupayaan menyelesaikan masalah global yang terlampau kompleks dalam sekelip mata.



Rajah 1.16 : Cadangan Ahli Parlimen dalam menyelesaikan masalah seharian



Rajah 1.17 : Cadangan Ahli Parlimen dalam menyelesaikan masalah seharian

1.5.2 Jenis AI Berdasarkan Fungsi Kerja

Berdasarkan fungsi dan cara pemprosesannya, AI boleh dikelaskan kepada empat jenis:

i. **Mesin Reaktif (*Reactive Machines*):** Jenis AI paling asas yang tiada fungsi memori. Ia tidak boleh menyimpan data masa lalu untuk digunakan dalam membuat keputusan hari ini. Ia hanya melihat keadaan semasa dan bertindak balas terus (contoh: program komputer catur *IBM Deep Blue* yang menganalisis kedudukan buah catur di atas papan pada waktu tersebut sahaja).

ii. **Memori Terhad (*Limited Memory*):** AI yang boleh menyimpan dan menggunakan data masa lalu untuk jangka masa pendek bagi memperbaiki prestasinya. Ini adalah teknologi yang membolehkan sistem autopilot kenderaan otonom memproses data kelajuan kenderaan lain, jarak jalan, dan papan tanda lalu lintas secara masa nyata untuk mengawal pemanduan.

iii. **Teori Minda (*Theory of Mind*):** Tahap AI konsep(masih belum wujud) yang mampu memahami dan membaca emosi, keperluan psikologi, kepercayaan, dan proses pemikiran manusia untuk berinteraksi secara sosial seperti seorang manusia sebenar.

iv. **Kewarasan (*Self-Aware AI*):** Peringkat tertinggi AI yang mempunyai kesedaran diri, identiti diri, dan kewarasan emosi yang serupa atau melebihi manusia. Mesin jenis ini mempunyai kehendak dan pemikirannya sendiri. Ini masih bersifat teori spekulatif sains fiksyen.

1.5.3 Contoh Aplikasi AI dalam IoT (AIoT)

i. **Pengesanan Penyakit Ternakan Ikan (Akuakultur):** Menggunakan kamera IoT di dalam kolam ternakan yang digabungkan dengan algoritma pembelajaran mendalam (*deep learning*). AI akan menganalisis visual pergerakan dan keadaan kulit ikan untuk mengesan tanda-tanda awal penyakit atau luka, membolehkan penternak bertindak sebelum penyakit berjangkit ke seluruh kolam.

ii. **Pengesanan Beban Puncak untuk Utiliti Bangunan (*Ampotech*):** Platform utiliti yang menggunakan sensor AIoT untuk memantau penggunaan elektrik bangunan pejabat secara masa nyata. Algoritma AI akan meramal waktu beban puncak penggunaan elektrik dan secara automatik melaraskan sistem pendingin hawa voltan tinggi (*HVAC*) untuk menjimatkan kos tenaga tanpa mengorbankan keselesaan pengguna.

1.6 Mentafsir Kajian Kes Berkaitan Aplikasi IoT

1.6.1 Kajian Kes 1: Projek Pengurusan Bencana Banjir Ghana (UNESCO & Kerajaan Jepun)

i. **Latar Belakang:** Kawasan Volta River Basin di Ghana sering berhadapan dengan banjir besar akibat perubahan iklim yang ekstrem. Ini mengancam nyawa penduduk dan merosakkan sektor pertanian tempatan.

ii. **Penyelesaian IoT:** UNESCO, dengan pembiayaan dana daripada kerajaan Jepun, memasang perkakasan IoT berupa penderia paras air (*water level sensors*) dan stesen cuaca pintar di tapak perintis *Afienya*. Data bacaan paras air dan taburan hujan dihantar secara masa nyata menggunakan gelombang komunikasi radio ke platform *ThingsBoard* dan kemudiannya disepadukan dengan platform maklumat water global UNESCO, iaitu *IHP-WINS*.

iii. **Impak Pembangunan Mampan:** Komuniti setempat mendapat amaran awal banjir secara automatik melalui telefon mudah alih. Ini memberikan masa yang cukup untuk proses pemindahan penduduk ke tempat selamat, sekali gus menyokong secara langsung dasar SDG 6 (Air Bersih) dan SDG 13 (Tindakan Terhadap Iklim).

1.6.2 Kajian Kes 2: Pemantauan Kualiti Konkrit Tapak Pembinaan di Malaysia (CIDB & JKR)

i. **Latar Belakang:** Mengikut laporan Lembaga Pembangunan Industri Pembinaan Malaysia (CIDB), banyak kes struktur bangunan retak atau runtuh berpunca daripada penggunaan bahan konkrit berkualiti rendah, campuran air-simen yang salah, atau kegagalan kontraktor mematuhi prosedur bancuhan semasa proses penghantaran lori pembancuh.

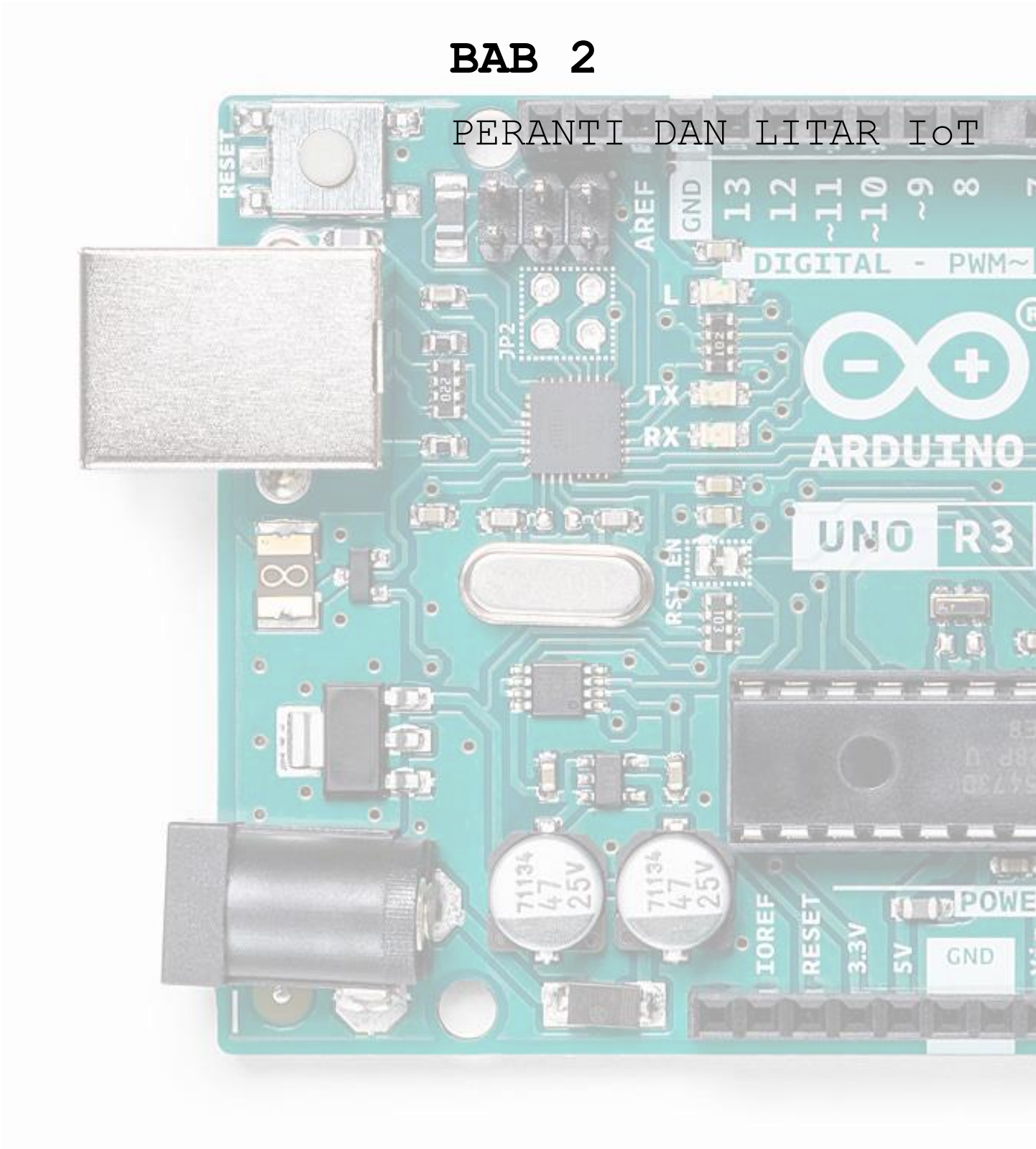
ii. **Penyelesaian AIoT:** Penderia AIoT khusus dimasukkan terus ke dalam tong dram lori pembancuh konkrit (*batching plant*) dan di tapak projek untuk memantau suhu, kelembapan, dan kadar kepekatan bancuhan konkrit secara masa nyata sepanjang proses pengerasan. Data ini dihantar ke platform awan dan dianalisis oleh algoritma AI untuk meramal kekuatan mampatan (*compressive strength*) konkrit pada hari ke-7 dan ke-28.

iii. **Impak Kecekapan Pengurusan:** Jurutera dan pihak berkuasa (seperti JKR) boleh mengesan ketidakpatuhan standard bancuhan serta-merta tanpa perlu menunggu ujian makmal manual yang memakan masa berminggu-minggu. Ini mengurangkan risiko kegagalan struktur bangunan, mengelakkan penipuan bahan binaan, dan menjamin keselamatan awam secara mampan.

KUIZ 2 : https://sites.google.com/kkkp.edu.my/ebook-internetofthings/home/kuiz-2

BAB 2

PERANTI DAN LITAR IoT



BAB 2: PERANTI DAN LITAR IoT (FOKUS: ARDUINO UNO)

Setelah anda memahami konsep teori dalam Bab 1, sekarang kita masuk ke bahagian teori yang menyentuh praktikal dengan **membina dan memprogramkan peranti IoT anda sendiri menggunakan platform pembelajaran terulung TVET—Arduino Uno!** Bab ini direka khas untuk melatih kemahiran amali anda daripada asas penyambungan perkakasan hinggalah kepada ketersambungan internet menggunakan modul Wi-Fi luaran.

2.1 Memahami Microcontroller (MCU)

2.1.1 Spesifikasi Perkakasan Arduino Uno sebagai Teras IoT

Dalam membina sebarang sistem terrbina dalam (*embedded system*) untuk aplikasi IoT, **Mikropengawal (MCU - *Microcontroller Unit*)** bertindak sebagai unit pemprosesan utama. Arduino Uno R3 menggunakan cip mikropengawal **ATmega328P** buatan *Microchip* (dahulunya Atmel).

Sebagai pengguna , kita wajib mengetahui perbandingan spesifikasi Arduino Uno dengan papan-papan pembangunan lain yang kerap digunakan di makmal seperti dalam jadual di bawah.

Jadual 2.1 : Perbezaan antara modul mikro pengawal dan pemproses mikro.

Ciri / Spesifikasi	**Arduino Uno R3 (ATmega328P)**	**ESP8266 (NodeMCU ESP-12E)**	**ESP32 (DOIT DevKit V1)**
Data *bus width* CPU	8-bit	32-bit	32-bit
Voltan Pengendalian	5.0V (Lebih baik)	3.3V	3.3V
Kelajuan Jam CPU	16 MHz	80 - 160 MHz	240 MHz (Dual-Core)
Memori SRAM	2 KB	80 KB	520 KB
Memori *Flash* Storan	32 KB	4 MB	4 MB
Komunikasi Wi-Fi	Tiada (Perlu modul ESP-01)	Terbina Dalam	Terbina Dalam

Resolusi Pembacaan ADC	10-bit (0 - 1023)	10-bit (Maksimum 1 pin)	12-bit (0 - 4095)
Arus Maksimum per Pin	20 mA - 40 mA	12 mA	12 mA - 40 mA

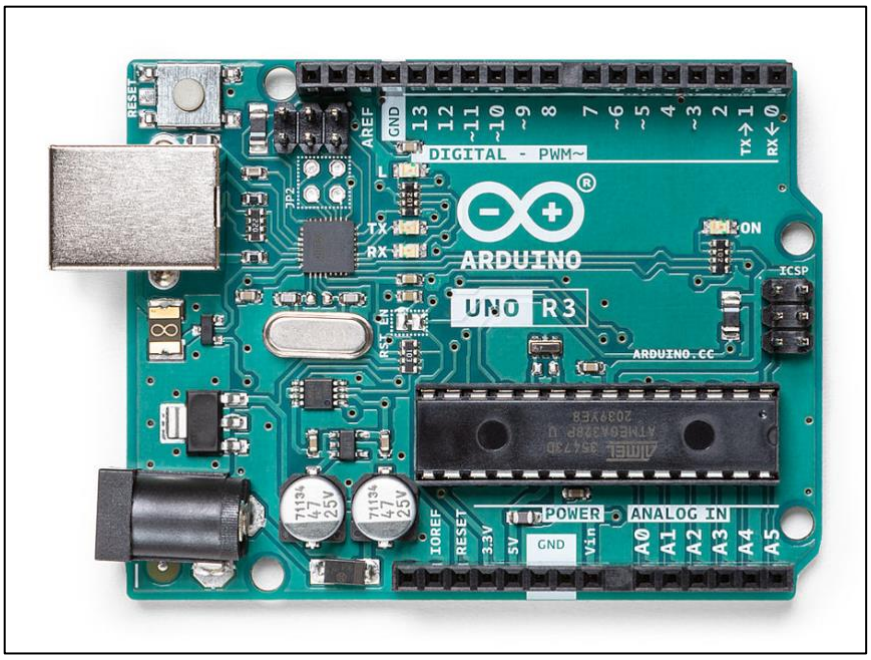
Analogi: Bayangkan CPU itu sebagai seorang "Chef", memori SRAM sebagai "meja memotong sayur", dan Memori Flash sebagai "kabinet menyimpan buku resipi".

> **Arduino Uno R3** ialah sebuah dapur praktikal bertaraf *heavy duty*! Walaupun dapurnya kecil dan chefnya bekerja pada kelajuan sederhana (16 MHz), dapurnya sangat lasak. Jika anda tersalah cucuk wayar (voltan 5V terkena pin isyarat secara tidak sengaja), litar Arduino Uno tidak akan terbakar dengan mudah. Ini sangat membezakannya dengan NodeMCU atau ESP32 yang sangat sensitif dan mudah rosak (*burn*) sekiranya menerima voltan lebih daripada 3.3V. Oleh itu, Arduino Uno ialah **papan latihan amali terbaik bagi pemula** sebelum mereka berpindah ke cip 3.3V yang lebih rumit.



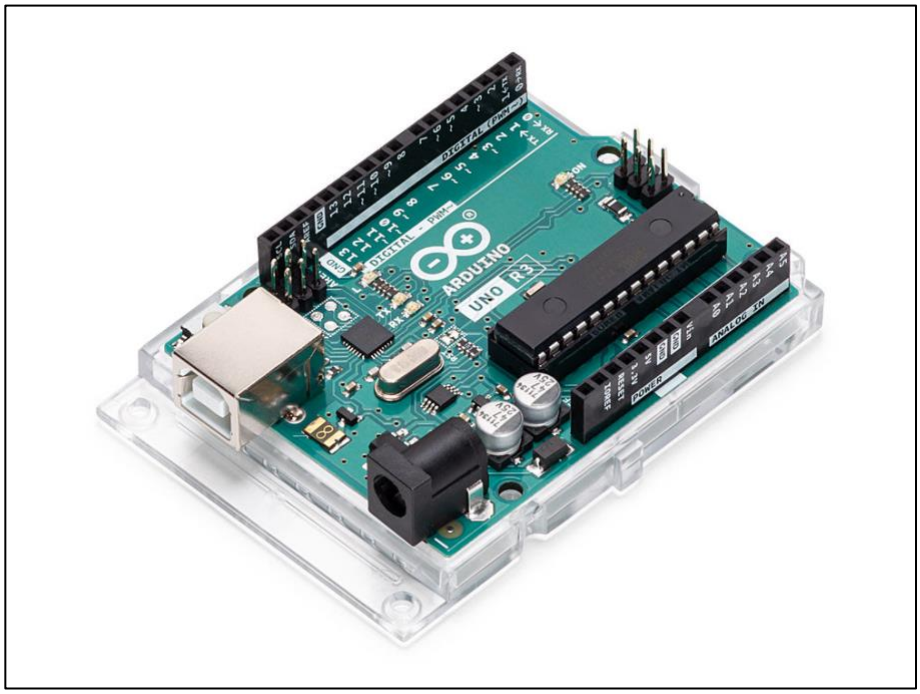
Scan QR ini menggunakan kamera anda untuk askes ke objek menggunakan *Augemented Reality*

https://asblr.com/sQDw7bW



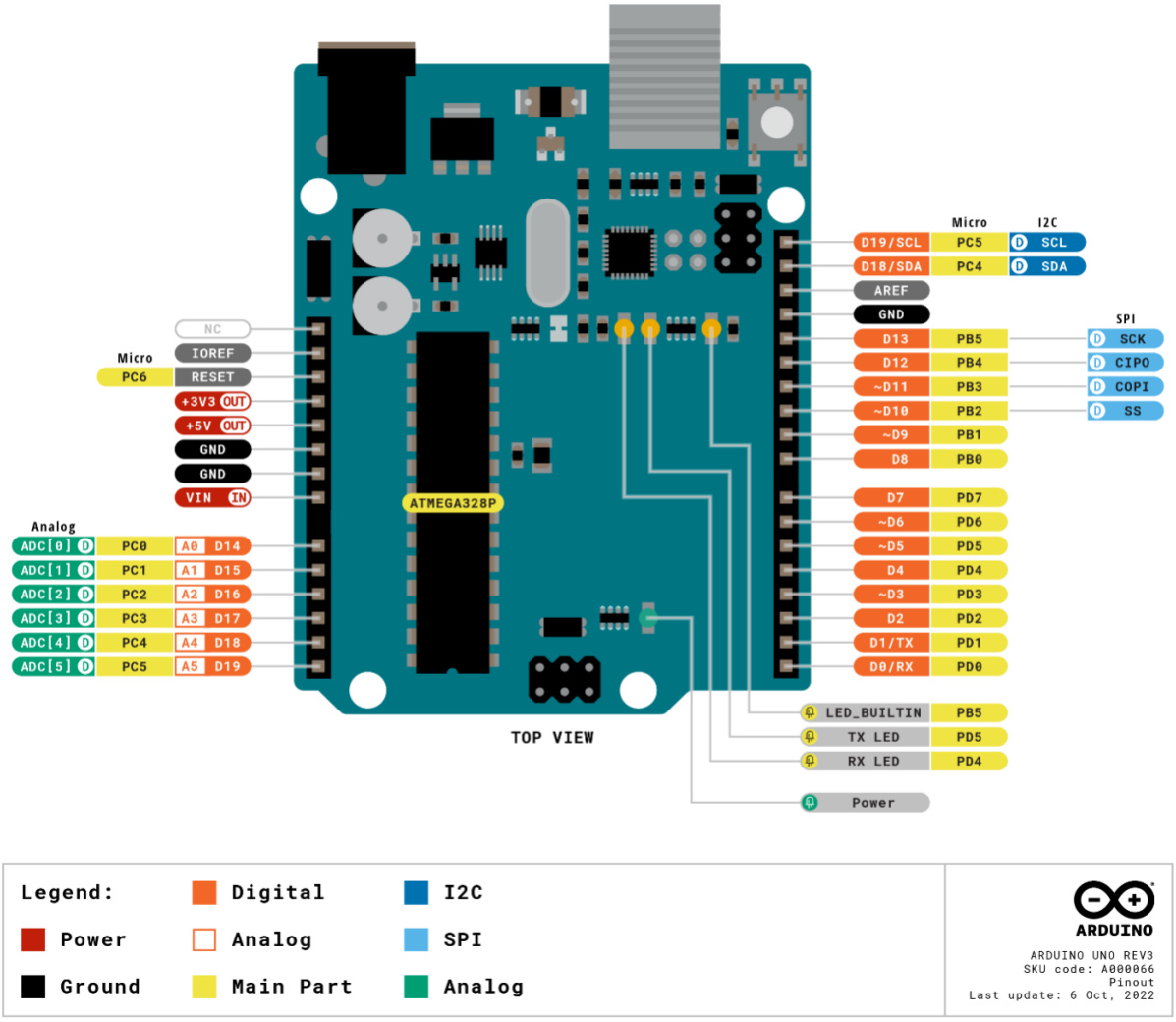
Rajah 2.1: Pengawal Mikro Arduino Uno

https://store.arduino.cc/products/arduino-uno-rev3



Rajah 2.2: Arduino Uno dari sudut pandangan yang berbeza

https://store.arduino.cc/products/arduino-uno-rev3



Rajah 2.3 : Susunan pin pada Arduino Uno

https://store.arduino.cc/products/arduino-uno-rev3

Arduino UNO R3 merupakan mikropengawal yang dioperasikan menggunakan pemproses ATMEL ATmega 328P yang mempunyai 20 pin yang boleh digunakan untuk sambung input seperti *sensor* , LED, motor dan peranti lain. Setiap pin punya fungsi tertentu seperti di bawah.

a. Pin Digital (0–13)

Pin ini menggunakan masukan dan keluaran digital dalam bit 1024 untuk mentafsir masukan dan keluaran . Ia boleh digunakans ebagai input atau output mengikut konfigurasi pada pengaturcaraan

- **Pin 0-1**: RX/TX (komunikasi dengan komputer)
- **Pin 2-13**: Guna untuk sensor digital, buzzer, relay, LED, button

b. Pin Analog (A0–A5) 🟡

Pin analog membaca nilai masukan pada tafsiran 256 bit. Biasanya digunakan pada sensor analog yang lebih mudah dipasang

- Pemasangan untuk penderia ringkas seperti Perintang Peka Cahaya atau *Light Dependent Resistor* (LDR), *potentiometer* dan pelbagai penderia yang lain yang memberikan bacaan analog
- Digunakan untuk membaca data berterusan

c. Pin KUASA (*Power*) 🔴

Memberikan kuasa kepada Arduino untuk beroperasi dan kuasa kepada peranti masukan atau keluaran yang lain melalui *DC jack* dengan kadar 5volt atau melalui pin USB 2.0 jenis B

- **5V**: Pin input kuasa untuk output yang kecil seperti LED atau kuasakan papan /*shield* lain.
- **3V3**: Pin input kuasa yang lebih rendah untuk sensor dan modul lain.
- **GND**: *Ground* (rujukan 0V) , penting untuk melengkapkan sebuah litar.
- **VIN**: Masukan kuasa dari luar menggunakan *DC jack*

d. Pin Komunikasi 🔵

Digunakan untuk komunikasi ke komputer atau ke mikro pengawal yang lain.

- **I2C** (SDA/SCL): Untuk kegunaan sensor, OLED, BME280 dan lain-lain.
- **SPI** (MISO, MOSI, SCK): Untuk SD card, beberapa sensor
- **IOREF + AREF**: Pin untuk rujukan kuasa pada litar

2.1.2 Perisian Microcontroller (MCU)

a. Persekitaran Pembangunan Bersepadu (IDE - *Integrated Development Environment*)

Setiap perkakasan memerlukan perisian yang merupakan platform membina pengaturcaraan. Aturcara yang ditulis merupakan kumpulan arahan atau *coding.* *Coding* pula ditulis menggunakan *firmware* yang dikenali sebagai **IDE**. Arduino IDE ialah sejenis perisian komputer yang menggabungkan semua alat pengaturcaraan yang diperlukan—seperti editor teks untuk menulis kod, *compiler* untuk menukar kod kepada bahasa mesin, serta tetingkap pengesan ralat (*debugger*)—ke dalam satu antara muka grafik yang mesra pengguna.

Sebelum anda bermula dengan Arduino IDE, anda perlu faham bahawa aturcara dibina menggunakan **bahasa C++** yang menggunakan kaedah pengaturcaraan berjujukan atau *sequential / procedural programming.* Satu lagi kaedah pengaturcaraan iaitu *object-oriented programmming* atau pengaturcaraan berorientasikan objek yang tidak dikupas dalam buku ini.

Sepintas untuk aturcara

NOTA: Arahan Asas Aturcara Arduino

a. Struktur Asas Program (*Sketch*)

Setiap program Arduino (dipanggil **"sketch"**) mesti ada 2 fungsi utama iaitu *SETUP* atau aturcara yang menentukan input, output, fungsi pin dan sebagainya. Manakala yang kedua ialah *LOOP* atau tempat arahan dilaksanakan.

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:
}
```

Selepas ini, kod akan ditulis dalam format `Andale Mono` untuk anda mengecam ia adalah aturcara, manakala komen akan ditulis dalam format `Abadi` untuk anda mengenali bahawa ia adalah komen. Seperti ditunjukkan di bawah.

```
void setup() {
  pinMode (12, OUTPUT)      // Kod di sini HANYA jalan SEKALI bila board start/reset
                            //  tempat untuk sediakan bahan2 nak masak
}

void loop() {
 digitalWrite(12, HIGH);    //  Kod di sini jalan BERULANG-ULANG selama-lamanya
                            //  Kaedah memasak
}
```

Rajah 2.4 Penerangan lanjut *function* asas C++ dalam Arduino



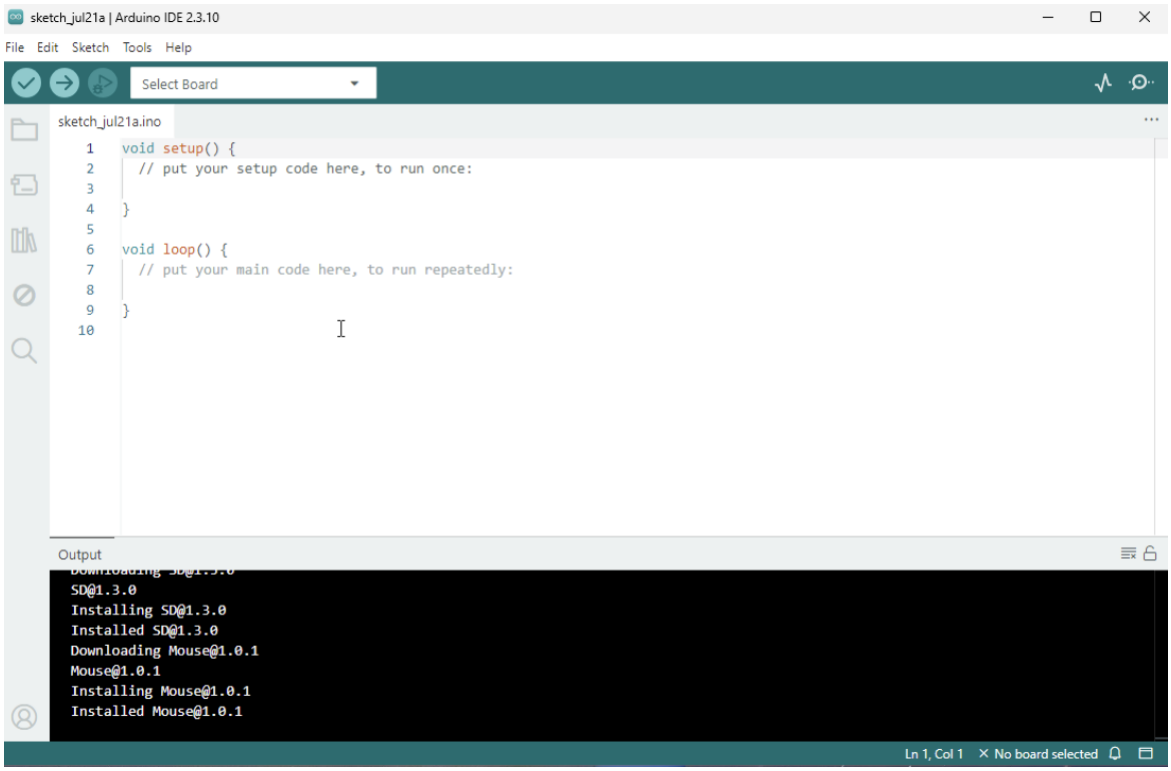
Rajah 2.5 : Function asas dilaksanakan dalam *sketch* iaitu Arduino IDE

Jadual 2.1 : Penerangan *function* asas dalam Arduino IDE

Istilah BM	**English**	**Maksud**
Fungsi	*Function*	Blok kod yang buat satu tugasan tertentu
Setup	*Setup*	Bahagian permulaan (sekali sahaja)
Loop	*Loop*	Bahagian berulang (gelung)
Sketch	*Sketch*	Nama fail program *Arduino*



Rajah 2.6 : Antaramuka Arduino yang asas
https://docs.arduino.cc/software/ide-v2/tutorials/getting-started-ide-v2/

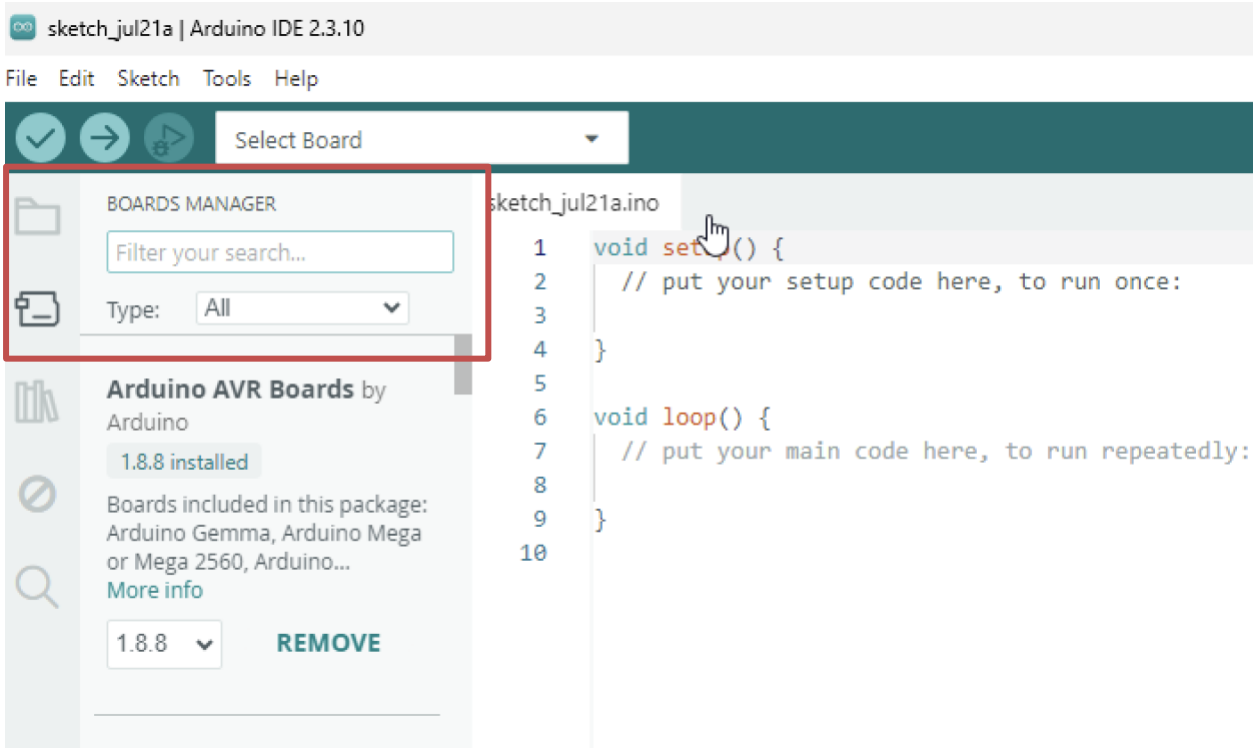


Rajah 2.7 : Arduino bersama tetingkap output

b. IDE Tools (Alatan Pengaturcaraan Penting)

Di dalam Arduino IDE, terdapat beberapa menu alatan teras yang wajib dikuasai semasa waktu amali:

1. ***Board Selector (Tools > Board):*** Digunakan untuk memilih jenis papan yang sedang disambungkan ke komputer. Untuk amali kita, kita sentiasa memilih **"Arduino Uno"**.
2. ***Port Selector (Tools > Port):*** Tempat di mana kita memilih port komunikasi fizikal (COM Port) yang digunakan oleh kabel USB komputer kita untuk berhubung dengan Arduino.
3. ***Library Manager (Sketch > Include Library > Manage Libraries):*** Tempat di mana kita memuat turun kod pustaka sedia ada secara percuma untuk mengurangkan beban menulis kod dari awal.
4. ***Serial Monitor:*** Tetingkap paparan teks yang menunjukkan data komunikasi antara Arduino Uno dan komputer secara dua hala melalui kabel USB.
5. ***Serial Plotter:*** Berfungsi menukarkan data teks berbentuk angka daripada Arduino Uno kepada bentuk graf garisan dinamik secara masa nyata.



Rajah 2.8 : Arduino IDE bersama *library /boards manager*

2.2 Menerangkan fungsi *sensor* (penderia) sebagai Peranti IoT

2.2.1 Kepentingan *Sensor* dalam Aplikasi IoT

Sensor (penderia) bertindak sebagai **pancaindera** bagi sistem IoT. Peranan utamanya adalah untuk mengesan dan mengukur perubahan parameter fizikal dalam dunia nyata dan menukarkan maklumat fizikal tersebut kepada isyarat elektrik (sama ada dalam bentuk voltan analog atau isyarat digital) yang boleh difahami oleh Arduino Uno.

2.2.2 Jenis-jenis Sensor yang Kerap Digunakan di Bengkel

a. **Sensor Suhu & Kelembapan (DHT11):** Menghasilkan output isyarat digital secara bersiri (*single-bus*) yang mengandungi data suhu dan kelembapan udara persekitaran.[8]
b. **Sensor Cahaya (LDR - Light Dependent Resistor):** Sejenis perintang berubah yang nilai rintangannya menurun apabila menerima cahaya yang tinggi. Ia mengeluarkan isyarat voltan analog.[8]
c. **Sensor Jarak Ultrasonik (HC-SR04):** Beroperasi dengan memancarkan gelombang bunyi frekuensi tinggi dan mengukur masa pantulan gelombang tersebut kembali ke sensor.
d. **Sensor Kelembapan Tanah (Soil Moisture Sensor):** Mengukur kekonduksian elektrik tanah berdasarkan kandungan air di dalam tanah.[8] Ia mengeluarkan isyarat analog.[8]
e. **Sensor Pergerakan (PIR - Passive Infrared):** Mengesan pergerakan haba tubuh manusia. Ia mengeluarkan isyarat digital HIGH (voltan 5V) apabila mengesan pergerakan.

2.2.3 Keperluan Kuasa Sensor dan Sambungan ke Arduino I/O

SENSOR 5V NATIVE ——> ——> PIN I/O ARDUINO UNO (Native 5V)
(Sangat Selamat, Tidak Memerlukan Logic Level Converter!)

a. **Paras Voltan Logik 5V:** Arduino Uno beroperasi pada voltan logik **5.0V**. Ini adalah satu kelebihan besar berbanding ESP32 atau Raspberry Pi. Kebanyakan *sensor* yang dijual di pasaran direka asli untuk menggunakan bekalan kuasa 5V. Apabila menggunakan Arduino Uno, anda boleh menyambungkan isyarat output *sensor* 5V (seperti PIR atau Ultrasonik) terus ke pin digital Arduino Uno tanpa bimbang tentang kerosakan voltan lampau.
b. **Pin Analog A0 hingga A5:** Arduino Uno mempunyai 6 pin input analog khusus. Pin-pin ini disambungkan ke litar ADC dalaman ATmega328P yang mempunyai resolusi 10-bit.
c. **Perintang Pukal Dalaman (*INPUT_PULLUP*):** Untuk mengelakkan pin input digital Arduino berada dalam keadaan "terapung" (*floating state*) akibat gangguan elektrostatik,

kita boleh menggunakan arahan pinMode(pin, INPUT_PULLUP). Ini mengaktifkan perintang pemukal 20k Ohm dalaman yang menarik voltan pin ke arah 5V (HIGH) secara automatik sehinggalah butang tekan ditarik ke GND (LOW).

2.3 Menerangkan Fungsi Aktuator sebagai Peranti IoT

2.3.1 Motor DC dan Relay dalam Aplikasi IoT

a. Geganti (Relay) sebagai Suis Elektromagnetik

Geganti bertindak as suis kawalan jauh fizikal. Pin GPIO Arduino Uno hanya boleh mengeluarkan arus maksimum 40mA sahaja, tidak cukup untuk menghidupkan kipas atau pam air elektrik. Geganti menyelesaikan masalah ini dengan menggunakan gelung elektromagnet voltan rendah (5V) untuk menarik sesentuh mekanikal bagi melengkapkan litar elektrik berasingan yang menggunakan voltan tinggi (contoh: AC 240V atau DC 12V). Sesentuh mekanikal geganti mempunyai tiga terminal utama:

i. **COM (*Common*):** Terminal punca bekalan elektrik utama yang dikawal.
ii. **NO (*Normally Open*):** Sentiasa terbuka (terputus sambungan dengan COM) secara lalai. Ia hanya akan bersambung dengan COM apabila geganti diaktifkan.
iii. **NC (*Normally Closed*):** Sentiasa tertutup (tersambung dengan COM) secara lalai.

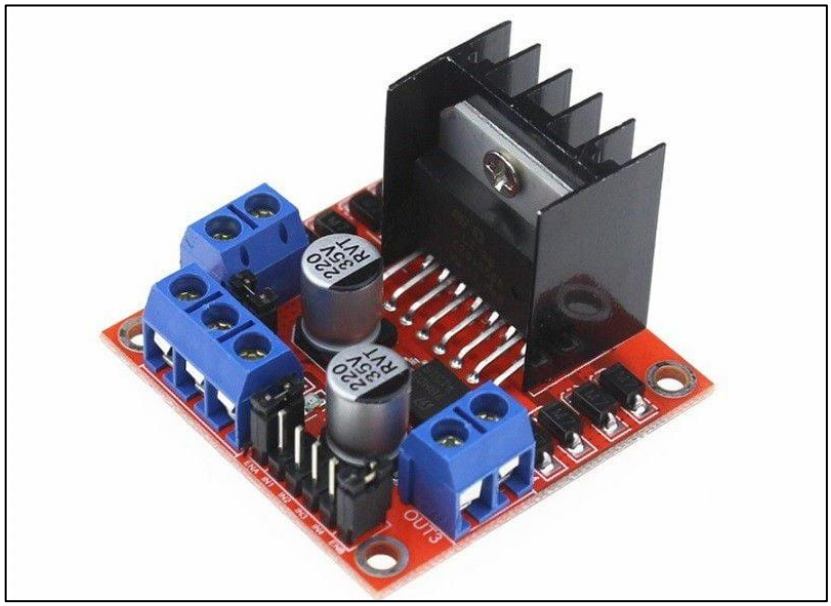
Tip di Bengkel:
Modul Geganti Logik *Active-Low*

Kebanyakan modul geganti 5V yang digunakan di bengkel amali TVET adalah jenis ***Active-Low***. Ini bermakna geganti akan aktif (suis tertutup/ON) apabila pin kawalan Arduino Uno diberi isyarat *LOW* (0V), dan akan padam (suis terbuka/OFF) apabila diberi isyarat *HIGH* (5V).

b. Motor DC dan Pemacu L298N (*H-Bridge*)

Motor DC menukarkan tenaga elektrik kepada tenaga mekanikal putaran. Untuk mengawal arah putaran (ke hadapan atau ke belakang) serta kelajuan motor DC dengan selamat, kita menggunakan **Modul Pemacu Motor L298N**. Modul ini mengandungi litar Jambatan-H (*H-Bridge*) yang mengawal aliran arah arus ke motor berdasarkan isyarat logik pemasaan PWM

(*Pulse Width Modulation*) dari Arduino Uno. Pemacu ini boleh dirujuk sebagai pemacu yang lengkap untuk mereka yang baru mula belajar tentang *motor driver* yang ringkas



Rajah 2.9 : pemacu motor L298N

2.4 Menetapkan Perisian IDE dan Peranti IoT

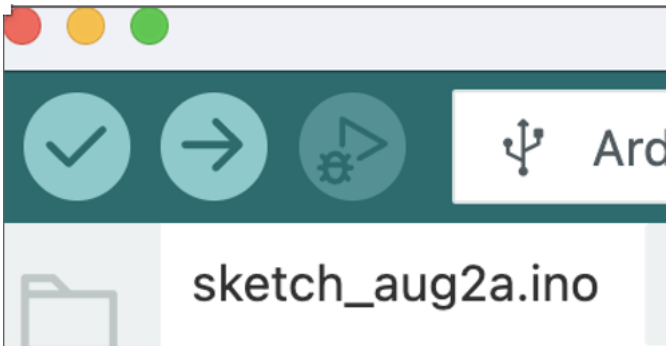
2.4.1 Menyediakan Perisian IDE untuk Arduino Uno

Salah satu sebab utama mengapa Arduino Uno sangat popular adalah kerana proses penyediaannya yang sangat ringkas dan tidak memerlukan pemasangan fail sokongan luaran seperti JSON untuk mikro pengawal yang lain, Selepas meluncurkan Arduino IDE, sila lakukan langkah-langkah di bawah:

a. **Sambungkan Arduino Uno:** Gunakan kabel USB jenis-B (kabel pencetak standard) untuk menyambungkan papan Arduino Uno ke port USB komputer anda.
b. **Pemacu (*Driver*) CH340 untuk Papan Klon:** Anda wajib memasang pemacu bagi cip **CH340** secara manual. Sebaik sahaja dipasang, Port COM maya akan muncul di dalam pengurus peranti (*Device Manager*).
c. **Tetapan Board pada IDE:** Buka Arduino IDE, pergi ke **Tools > Board > Arduino AVR Boards** dan pilih **"Arduino Uno"**.
d. **Tetapan Port pada IDE:** Pergi ke **Tools > Port** dan pilih port COM yang aktif (contoh: COM3 atau COM4).

2.4.2 Membina Sambungan, Menyusun Semula, dan Memuat Naik Kod

a. **Penyusunan Semula (*Compiling*):** Sebelum memuat naik kod, klik butang ***"Verify"*** (simbol tanda check ✓) di dalam Arduino IDE. IDE akan menyemak sintaks penulisan kod anda dan menterjemahkannya kepada bahasa mesin (fail perduaan)



Rajah 2.10 : butang *Verify* dan *Upload* dalam Arduino IDE

b. **Memuat Naik (*Uploading*):** Klik butang ***"Upload"*** (simbol anak panah ke kanan →). IDE akan menghantar kod program tersebut terus ke dalam memori Flash ATmega328P. Berbeza dengan ESP32, proses muat naik pada Arduino Uno adalah **100% automatik** tanpa memerlukan anda menekan sebarang butang "BOOT" secara manual!

2.5 Menetapkan Input dan Output dalam Aplikasi Sebenar dengan Platform IoT

Dibawah ialah aturcara yang membuatkan LED di kaki 13 berkelip secara berkala setiap 1 saat. Maka kita perlu tetapkan kaki 13 di Arduino UNO sebagai keluaran. Supaya mikropengawal akan membekalkan isyarat TINGGI dan RENDAH mengikut kehendak pengguna.

```
void setup() {
  pinMode(13, OUTPUT);      // set pin 13 sebagai output,sebagai contoh LED letak di pin 13, maka LED ialah output
}

void loop() {
  digitalWrite(13, HIGH);   // LED ON
  delay(1000);              // tunggu 1 saat (1000 milisaat),LED akan ON selama 1 saat
  digitalWrite(13, LOW);    // LED OFF
  delay(1000);              // tunggu 1 saat ,LED akan OFF selama 1 saat
}
```

(Ini contoh klasik "Blink" — lampu LED berkelip)

Jadual 2.2 Penerangan setiap baris arahan dalam aturcara pertama

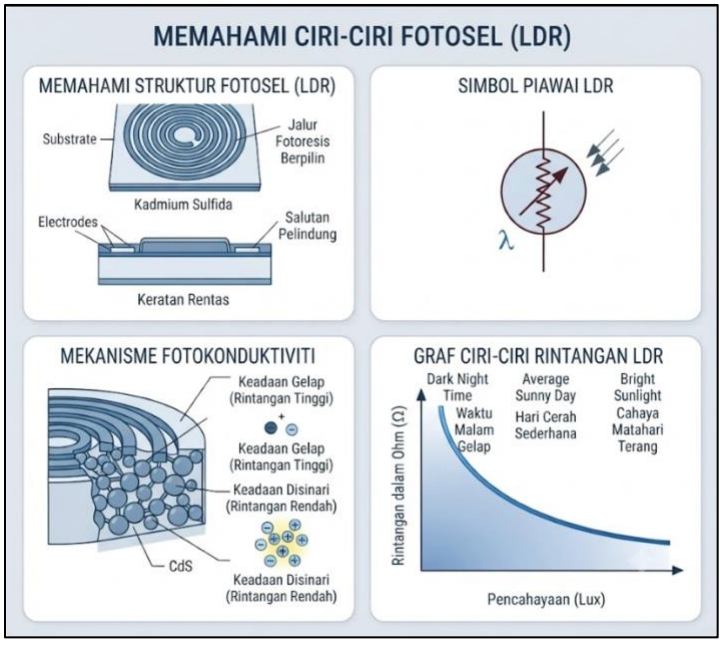
Istilah	English	Maksud
TINGGI	High	Signal ON / 5V
RENDAH	Low	Signal OFF / 0V
KELUARAN	Output	Pin hantar signal keluar (contoh: ke LED)
MASUKAN	Input	Pin terima signal masuk (contoh: dari suis)
mili saat (ms)	millisecond	1/1000 saat

2.5.1 Membaca Data dari Peranti Input (Fasa Tempatan)

Di dalam litar Arduino Uno, isyarat voltan analog daripada *sensor* akan dibaca oleh pin analog **A0**. Litar ADC 10-bit pada Arduino Uno akan menukarkan voltan fizikal (0V - 5V) kepada nilai integer digital berjulat antara 0 hingga 1023. Rumus matematik penukaran voltan analog kepada nilai ADC 10-bit Arduino Uno ialah:

$$D_{out} = \frac{V_{in}}{V_{ref}} \times 1023$$

Di mana $V_{ref} = 5\text{V}$, dan D_{out} ialah nilai integer digital yang kita perolehi melalui fungsi *analogRead().*



Di dalam rajah di sebelah ditunjukkan kendalian bagi contoh sensor termudah iaitu Perintang Peka Cahaya atau *Light Dependant Resistor (LDR)*

Rajah 2.11 : operasi dan kendalian LDR

Untuk kes seperti LDR di bawah , reaksi bahan dalam sensor itu akan memberikan rintangan pada voltan yang melaluinya lalu pemproses boleh menilai perubahan yang berlaku samada cahaya terang atau keadaan gelap sedang berlaku. Jika nilai voltan menjadi rendah kerana rintangan tinggi , maka adalah gelap. Jika keadaan sebaliknya maka pemproses tahu keadaan adalah cerah. Sila rujuk graf yang ditunjukkan seperti rajah di sebelah.

2.5.2 Menghubungkan Arduino Uno ke Platform IoT (*Blynk* Cloud)

Satu-satunya kelemahan Arduino Uno ialah ia **tidak mempunyai cip Wi-Fi terbina dalam**. Bagi membolehkan Arduino Uno berkomunikasi dengan platform *BlynkCloud*, kita perlu menyambungkannya bersama sebuah modul Wi-Fi luaran yang kos rendah iaitu **ESP-01 (ESP8266)**.

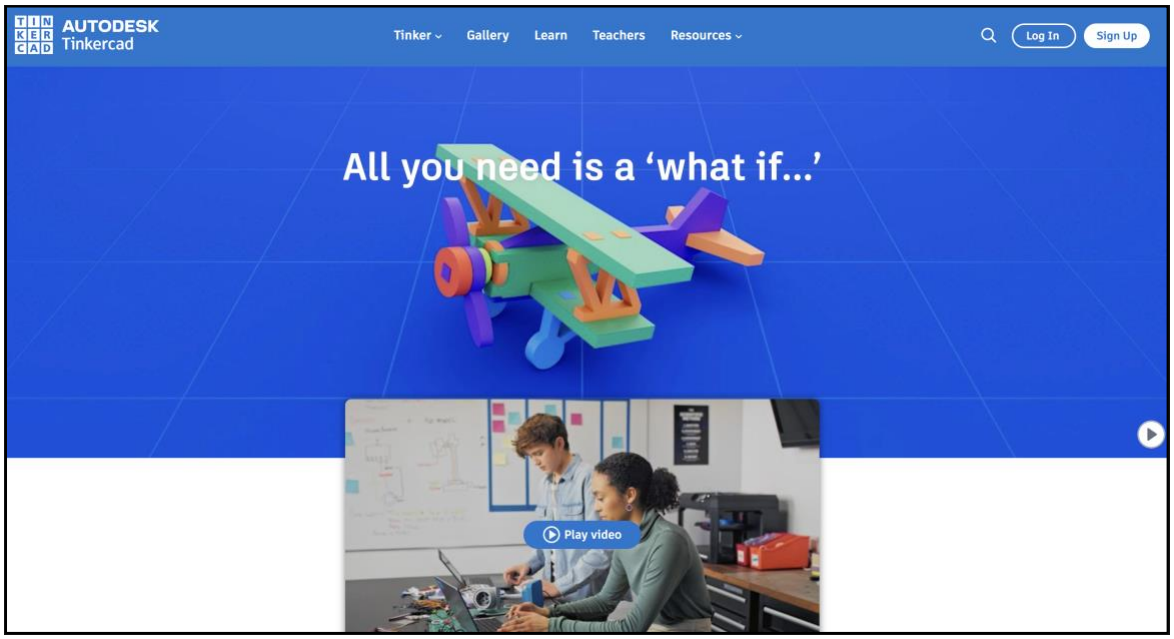
Di dalam persediaan ini, Arduino Uno akan bertindak sebagai pengumpul data sensor dan pembuat keputusan utama, manakala modul ESP-01 bertindak sebagai "modem Wi-Fi" luaran.

ARDUINO UNO (Master) ———> ESP-01 (Wi-Fi Modem) ———> BLYNKCLOUD

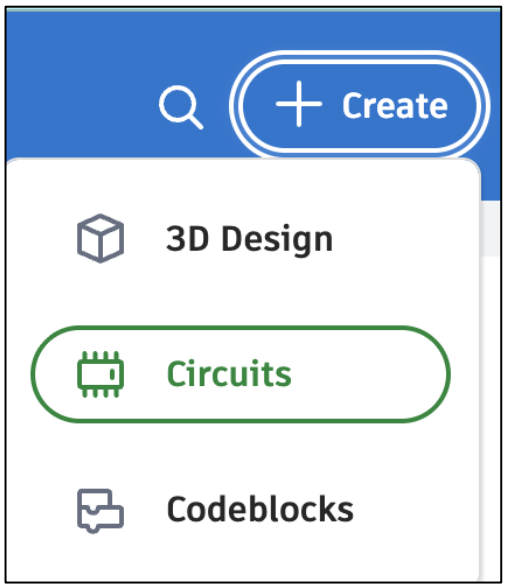
Untuk pemahaman praktikal dengan lebih baik dan tersasar, kita akan gunakan simulasi untuk memahami sifat litar yang asas dahulu dan operasinya. Kita juga akan memahami kesan perubahan aturcara kepada *output* litar.

KHAS : kaedah pembinaan litar menggunakan simulasi

a. Layari lamanweb www.tinkercad.com .Login menggunakan akaun (*google* dan sebagainya) dan pilih *create* “CIRCUIT”



Rajah 2.11 : Antraramuka permulaan TinkerCAD oleh AutoDesk.
https://www.tinkercad.com/



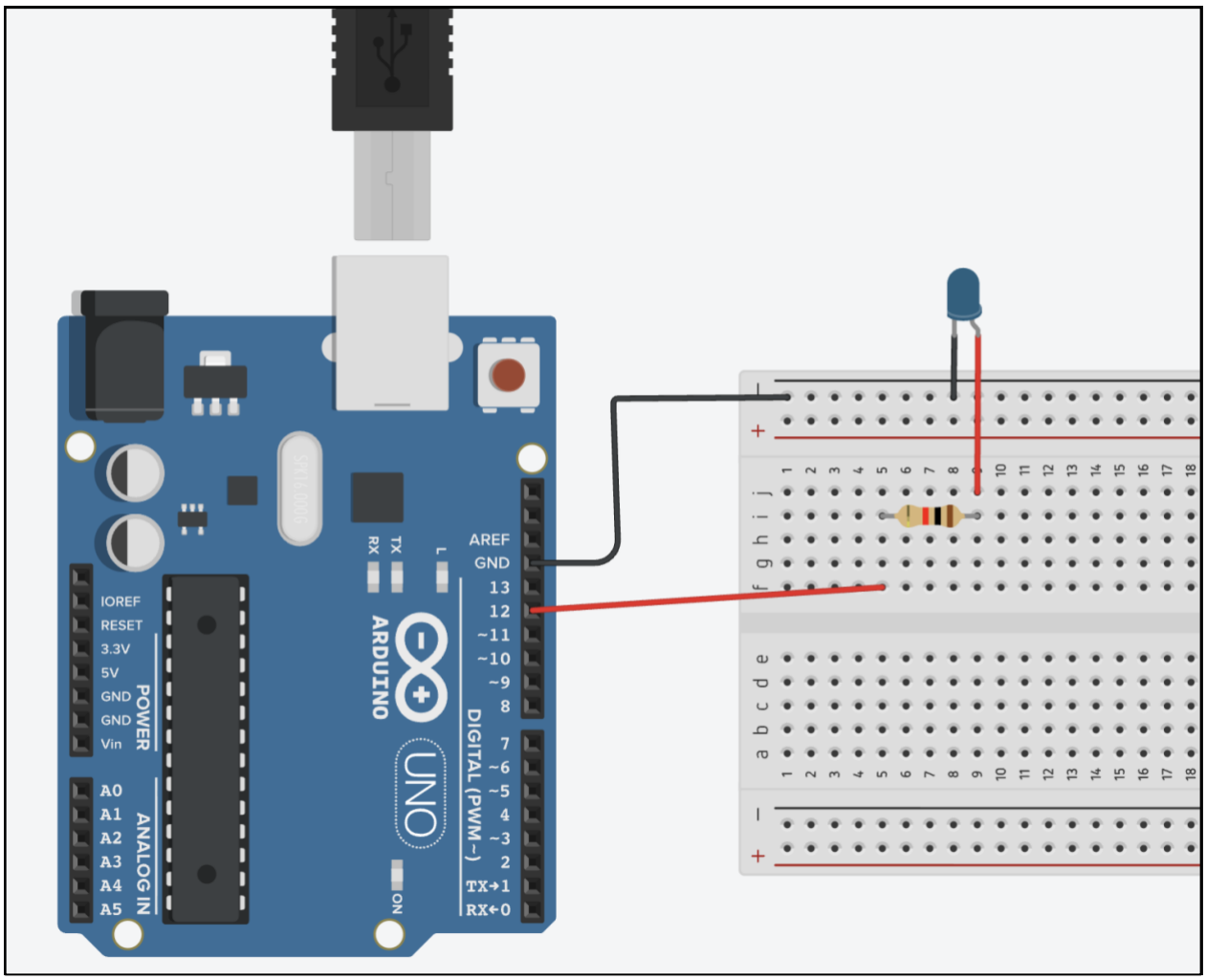
b. Masukkan komponen tersebut dan buat sambungan seperti di bawah:

i. Arduino UNO

ii. Perintang 1kOhm

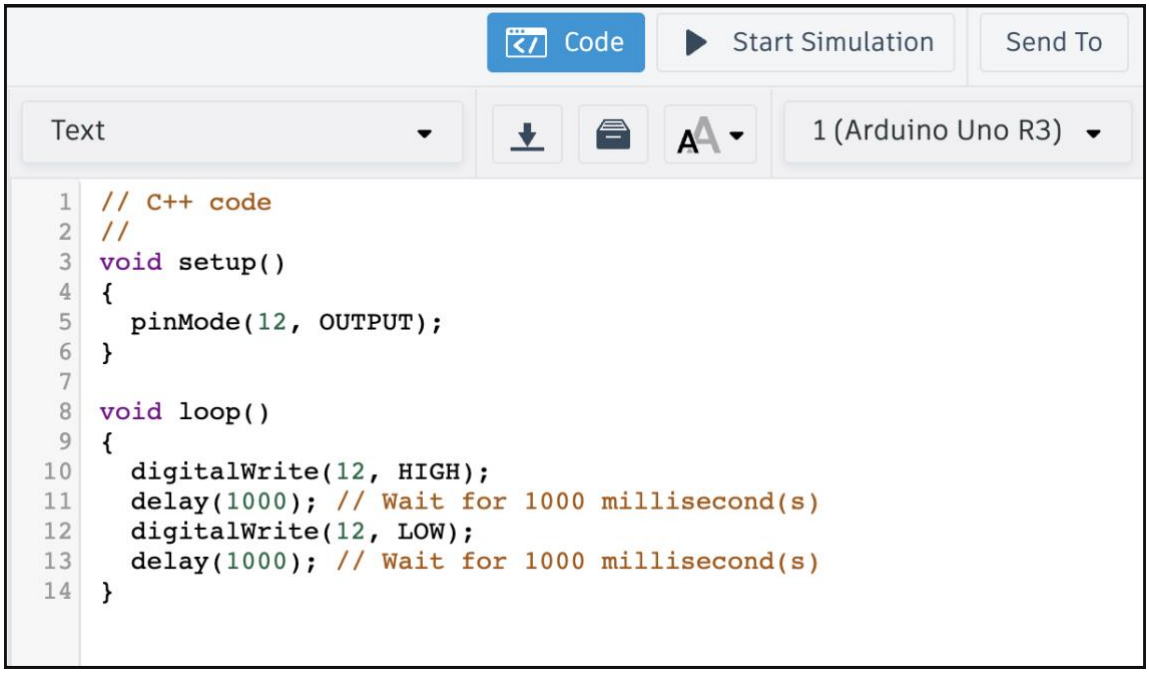
iii. LED (apa-apa warna)

iv. *Breadboard*



Rajah 2.12 : sambungan untuk LED berkelip menggunakan Arduino Uno

c. Masukkan kod seperti di bawah dan lihat hasilnya



Rajah 2.13 : Kod LED berkelip disalin dalam aplikasi TinkerCAD.

d. Berikut ialah penerangan mengenai kod yang dimasukkan. SOALAN: bolehkah anda ubah kod supaya LED berkelip lebih laju atau tambah lebih dari dua LED?

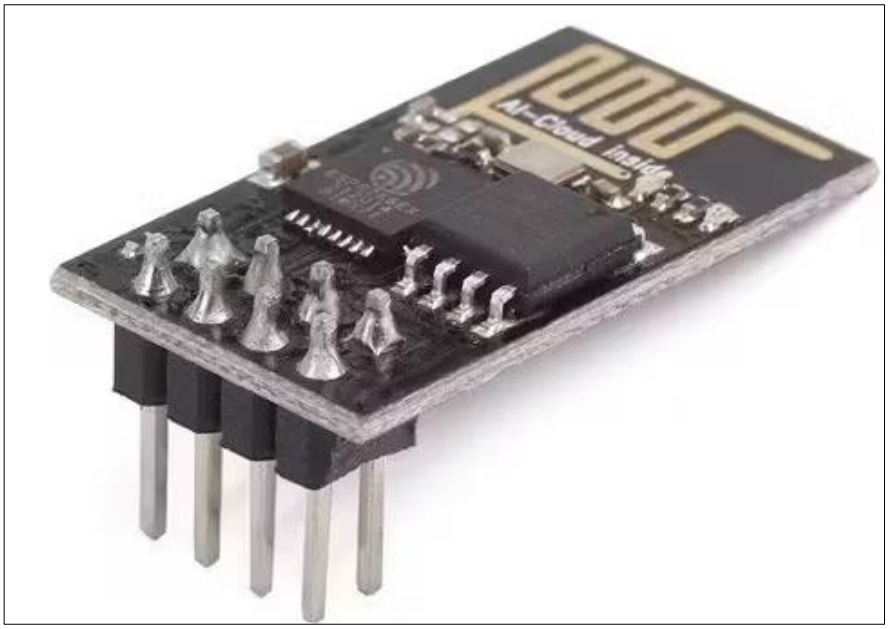
```
void setup() {
  pinMode(13, OUTPUT);     // set pin 13 sebagai output
}

void loop() {
  digitalWrite(13, HIGH);  // LED akan ON
  delay(1000);             // tunggu 1 saat (1000 milisaat)
  digitalWrite(13, LOW);   // LED akan OFF
  delay(1000);             // tunggu 1 saat (1000 milisaat)
}
```

Rajah 2.14 : Aturcara dan penerangan bagi projek LED berkelip

Tambahan: SOP Amali & Cabaran Voltan Logik 3.3V Modul ESP-01 (Tip Emas Pelajar)

Untuk menjadikan sebuah mikro pengawal Arduino Uno boleh menyertai Internet atau sebahagian daripada *Internet of Thing (IoT),* litar tersebut memerlukan modul tambahan iaitu ESP-01 . Rajah di bawah menunjukkan modul ESP-01 yang akan digunakan,



Rajah 2.15 : Modul ESP-01 yang menjadikan Arduino Uno boleh disambungkan ke Internet.

Satu kesilapan besar yang kerap merosakkan modul ESP-01 di dalam bengkel ialah **isu perbezaan voltan logik**. Arduino Uno menggunakan voltan isyarat **5V**, manakala modul ESP-01 hanya boleh menerima voltan logik maksimum **3.3V**.

a. **Pin Bekalan Kuasa VCC ESP-01:** Mesti disambungkan ke punca kuasa **3.3V** (bukan 5V!). Namun, litar kuasa 3.3V terbina dalam Arduino Uno tidak mempunyai arus yang cukup besar untuk memandu cip pemancar *Wi-Fi* ESP-01 secara stabil. Anda digalakkan menggunakan modul pembahagi kuasa khusus untuk ESP-01.
b. **Pin Rx (*Receive)* ESP-01:** Pin ini tidak boleh menerima isyarat 5V terus daripada pin Tx (*Transmit*) Arduino Uno. Anda **wajib memasang litar pembahagi voltan (*voltage divider*)** menggunakan dua perintang (contoh: 1k Ohm dan 2k Ohm) pada talian isyarat Tx Arduino Uno ke Rx ESP-01 bagi menurunkan voltan 5V kepada 3.3V untuk mengelakkan kerosakan pada litar.

Konfigurasi Rangkaian Menggunakan *SoftwareSerial*

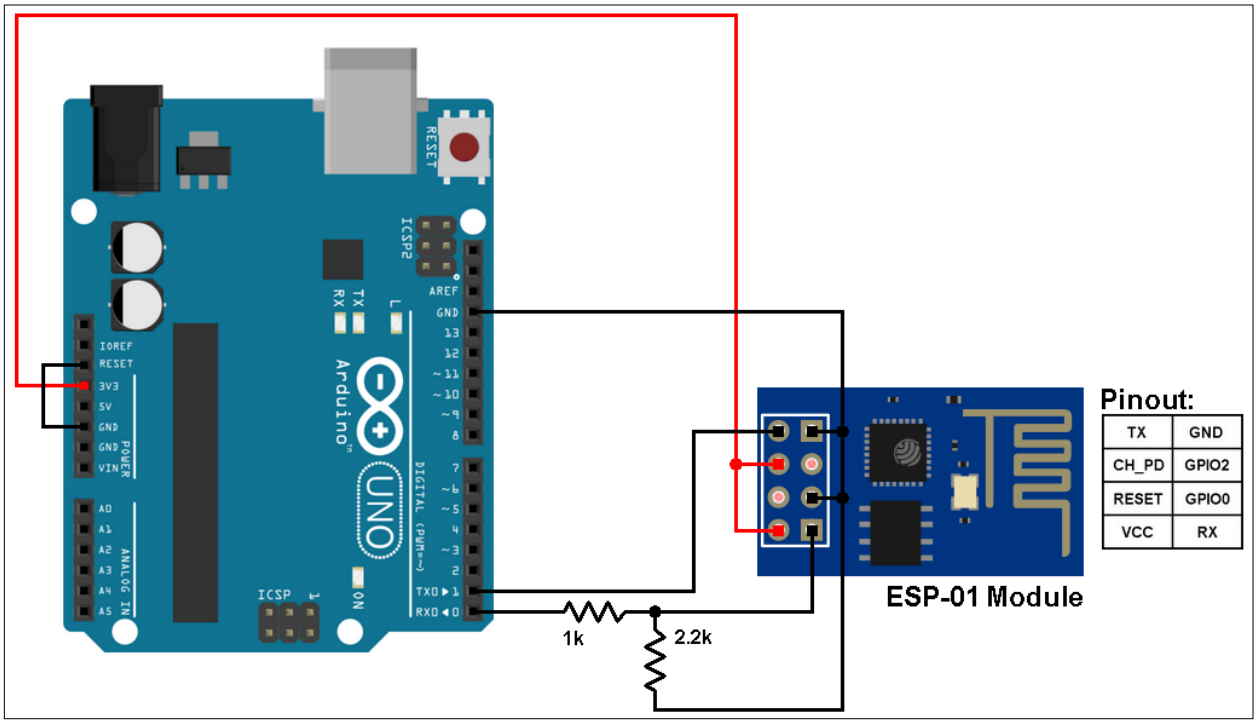
Untuk membolehkan Arduino Uno berkomunikasi dengan modul ESP-01 tanpa mengganggu sambungan kabel USB ke komputer, kita tambah menggunakan kod pustaka **SoftwareSerial.h**. *Library* ini menukarkan mana-mana pin digital biasa pada Arduino Uno menjadi pin komunikasi *Serial* (Rx/Tx). Dalam amali ini, kita menetapkan **Pin 8 sebagai Rx** dan **Pin 9 sebagai Tx**. Kelajuan kadar baud (*baud rate*) komunikasi serial ESP-01 ditetapkan pada **9600 baud** untuk kestabilan optimum.

2.5.3 Paparkan Data ke Peranti Output (Manual & *Blynk Cloud*)

Mari kita satukan keseluruhan sistem IoT! Kita akan membaca sensor DHT11, menghantar data tersebut ke platform BlynkCloud menggunakan modul ESP-01, dan membina butang suis pada aplikasi Blynktelefon pintar untuk mengawal modul geganti 5V pam air yang disambungkan ke **Pin Digital 3** Arduino Uno.

Penyediaan fail (*Library Setup*): Anda wajib memasang *library* ***Blynk*** serta **BlynkESP8266_Lib** melalui *Library Manager* di dalam Arduino IDE terlebih dahulu sebelum memulakan sesi pengaturcaraan.

Kod Program 2: Projek IoT Lengkap Arduino Uno + ESP-01 + BlynkCloud



Rajah 2.16 : Sambungan ESP-01 ke Arduino Uno untuk menjadikan ia *WiFi enabled*

2.6 Penyelesaian Masalah (*Troubleshooting*) dan Kesalahan Lazim di Bengkel

Apabila projek IoT Arduino Uno anda tidak berfungsi, jangan panik! Berikut adalah senarai semak penyelesaian masalah praktikal khusus untuk peranti Arduino Uno yang biasa berlaku di dalam bengkel:

2.6.1 Isu *Compiler* dan Muat Naik (*Compile & Upload*)

a. **Ralat: "Avoid using Pin 0 and Pin 1" / Serial Monitor Terpapar Tulisan Rosak**
   - *Punca:* Pin digital **D0 (Rx)** and **D1 (Tx)** pada Arduino Uno berkongsi talian fizikal yang sama dengan pengawal penukar USB komputer. Sekiranya anda menyambungkan sebarang komponen (seperti modul ESP-01 atau *sensor)* ke Pin 0 atau Pin 1 semasa proses memuat naik kod dijalankan, ia akan **mengganggu isyarat USB** dan menyebabkan ralat muat naik.
   - *Penyelesaian:* Sentiasa **cabut sebarang wayar** yang disambungkan ke Pin 0 dan Pin 1 semasa anda menekan butang *"Upload"* kod. Cara terbaik ialah gunakan SoftwareSerial pada pin digital lain (cth., Pin 8 & 9) untuk komunikasi luaran.

b. **Ralat: "Serial Port (COM) not found"**
   - *Punca:* Komputer tidak memasang pemacu CH340 Driver untuk papan klon.
   - *Penyelesaian:* Muat turun dan pasang CH340 Driver secara manual. Pastikan kabel USB yang digunakan menyokong talian data.

2.6.2 Isu Sambungan Rangkaian (*Network & Blynk*)

a. **Aplikasi BlynkMenunjukkan "*Device Offline"* secara Berterusan**
   - *Punca:* Arus elektrik dari pin 3.3V Arduino Uno tidak cukup kuat untuk membekalkan kuasa yang mencukupi kepada cip transiver Wi-Fi ESP-01 semasa ia menghantar isyarat data tinggi, menyebabkan ESP-01 kerap mengalami proses *restart* kendiri secara berulang.
   - *Penyelesaian:* Gunakan **bekalan kuasa luaran 3.3V** yang berasingan untuk menguasakan modul ESP-01. Pastikan anda **menyambungkan wayar Ground (GND) luaran tersebut sekali ke pin GND Arduino Uno** (*Common Ground*).

b. **Ralat Pembacaan Sensor "nan"**
   - *Punca:* Wayar penyambung longgar atau sensor DHT11 tidak mendapat bekalan kuasa yang cukup elok dari terminal *breadboard.*
   - *Penyelesaian:* Pastikan semua wayar jumper ditekan rapat ke dalam lubang *breadboard.* Periksa jika anda tersalah menyambung pin isyarat ke pin kuasa.

NOTA RUJUKAN

a. Sintaks (*Syntax)(*Cara Tulis Kod Yang Betul)

Syntax = peraturan tatabahasa dalam bahasa pengaturcaraan. Kalau silap *syntax*, kod akan ***error****.*

Simbol	Nama (English)	Fungsi
;	*Semicolon*	Wajib letak di **hujung setiap arahan** (macam noktah .)
{ }	*Curly braces*	Bungkus kumpulan arahan dalam satu fungsi/*loop*
( )	*Parentheses*	Tempat letak ***parameter*** dalam fungsi
//	*Double slash*	Komen (*comment*) — nota untuk manusia, tak dibaca mesin
/* ... */	*Block comment*	Komen banyak baris sekaligus

```
                        // Ini contoh komen — tak akan run
digitalWrite(13, HIGH);  // arahan wajib ada ";"
```

---

b. Variable (Pembolehubah)

Variable = "kotak" untuk simpan data/nilai yang boleh berubah.

```
int ledPin = 13;          // simpan nombor bulat (integer)
float suhu = 25.5;        // simpan nombor perpuluhan (decimal)
bool statusLampu = true; // simpan nilai true/false
```

Jenis (Data Type)	English	Contoh Nilai
int	Integer	13, -5, 100 (nombor bulat)
float	Float / Floating point	25.5, 3.14 (ada titik perpuluhan)
bool	Boolean	true / false (hidup/mati)
char	Character	'A', 'b' (satu huruf)
String	String	"Hello Dunia" (teks/ayat)

💡 **Tips :** *Variable* = macam label pada bekas plastik. Nama bekas (nama *variable*) kekal, tapi isi (nilai) boleh tukar.

---

c. *Parameter* (Nilai Dalam Fungsi)

Parameter = maklumat yang kita bagi kepada satu fungsi supaya ia tahu apa nak buat.

```
    pinMode(13, OUTPUT);
    ↑        ↑       ↑
//  parameter 1 (pin nombor)  parameter 2 (mod)
```

Fungsi	Parameter	Maksud (English)
`pinMode(pin, mode)`	pin, mode	Tetapkan pin sebagai INPUT / OUTPUT
`digitalWrite(pin, value)`	pin, value	Hantar signal HIGH (on) / LOW (off)
`digitalRead(pin)`	pin	Baca signal dari pin (HIGH/LOW)
`analogWrite(pin, value)`	pin, value (0–255)	Hantar signal analog (contoh: dim LED)
`analogRead(pin)`	pin	Baca nilai analog (0–1023) dari sensor
`delay(ms)`	ms (milisaat)	Berhenti/tunggu sekejap

---

d. Operator (Simbol Pengiraan & Perbandingan)

Operator	English	Contoh	Maksud
=	Assignment	x = 5	Letak nilai 5 dalam x
==	Equal to	x == 5	Semak jika x SAMA dengan 5
!=	Not equal	x != 5	Semak jika x TAK SAMA dengan 5
> <	Greater/less than	x > 5	Lebih besar / kecil
&&	AND (logical)	x>5 && y<10	Kedua-dua syarat mesti betul
||	OR (logical)	x>5 || y<10	Salah satu syarat cukup

---

e. Struktur Kawalan (*Control Structure)*

```
if (suhu > 30) {
  digitalWrite(fan, HIGH);   // jika suhu > 30°C, ON kipas
} else {
  digitalWrite(fan, LOW);    // kalau tidak, OFF kipas
}
```

Istilah	English	Maksud
if	If (conditional)	"Kalau..." — buat sesuatu jika syarat betul
else	Else	"Kalau tidak..."
for	For loop	Ulang sejumlah kali yang ditetapkan
while	While loop	Ulang selagi syarat betul

📌 Ringkasan Cepat *(Quick Recap)*

Istilah BM	English
Kod/Arahan	*Code / Statement*
Pembolehubah	*Variable*
Parameter	*Parameter*
Fungsi	*Function*
Gelung	*Loop*
Syarat	*Condition*
Sensor	*Sensor*
Pin masukan/keluaran	*Input/Output pin*



BAB 3

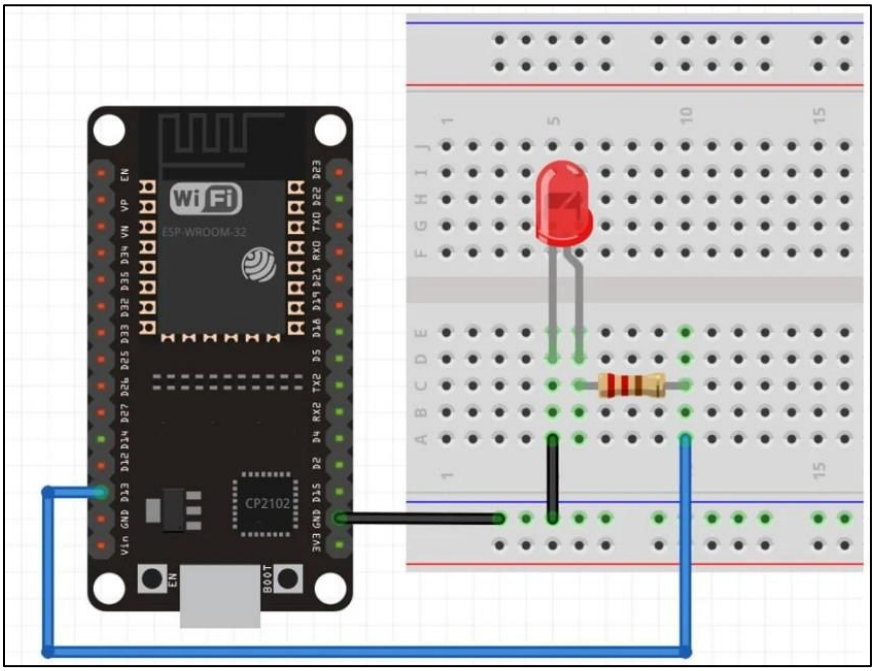
PEMBENTUKAN PROJEK ASAS DENGAN ESP32

Sumber foto: https://www.mecalux.com/blog/iiot-supply-chain

BAB 3: PEMBENTUKAN PROJEK ASAS DENGAN ESP32 (GABUNGAN BLYNK-IoT)

Selepas anda menguasai konsep dan amali asas menggunakan Arduino Uno dalam Bab 2, kini tiba masanya untuk kita melangkah setapak ke hadapan dengan menerokai mikropengawal yang paling berkuasa dan popular dalam industri IoT moden, iaitu **ESP32**. Di dalam Bab 3 ini, anda akan melakukan latihan amali bersepadu *BlynkIoT*: **mengawal status menyala LED luaran fizikal menggunakan pin maya (*Virtual Pin*) dari aplikasi telefon pintar**. Latihan ini akan membina asas kefahaman anda tentang cara mengintegrasikan perisian digital platform IoT dengan pin Output (GPIO) fizikal pada ESP32.

Dalam bahagian praktikal ini, kita akan membina projek menghidupkan LED yang ringkas. Kita akan membaca sensor DHT11 dan *sensor* kelembapan tanah sebagai input, memaparkan data tersebut di platform awan *Blynk*IoT, dan mengawal suis geganti untuk menghidupkan pam air fizikal sebagai output.



Rajah 3.10 : projek pertama pengujian ESP32



Scan QR ini menggunakan kamera anda untuk askes ke objek menggunakan *Augemented Reality*

Kredit : "ESP32 WROOM 32 - Low Poly Photorealistic" (https://skfb.ly/ptIMN) by wojteX is licensed under Creative Commons Attribution (http://creativecommons.org/licenses/by/4.0/).

https://asblr.com/h9DQ5y0

3.1 Pengenalan Ringkas Perkakasan ESP32

ESP32 ialah cip *System-on-Chip (SoC)* asas untuk kegunaan pelajar dan peminat IoT keluaran *Espressif Systems.* Walaupun Arduino Uno sangat lasak, ia tidak mempunyai keupayaan internet terbina dalam secara asli dan terpaksa bergantung kepada modul tambahan seperti ESP01. Modul ESP32 mengatasi kelemahan ini dengan mengintegrasikan modul **Wi-Fi 2.4 GHz** dan ***Bluetooth* (v4.2 / BLE)** secara terus di atas papan litar fizikalnya.

Satu perkara kritikal yang perlu diingatkan oleh pelajar TVET: **ESP32 beroperasi secara mutlak pada voltan logik 3.3V**. Jangan sesekali menyambungkan isyarat voltan 5V terus ke pin GPIO ESP32 untuk mengelakkan kerosakan akibat voltan lampau (*over voltage*)

3.2 Memahami Pin GPIO ESP32

Pin GPIO (*General Purpose Input/Output*) ialah pin fizikal yang boleh dikonfigurasikan melalui kod program untuk berfungsi sama ada sebagai input mahupun output digital. Dalam amali ini, kita menetapkan **GPIO 5** sebagai output digital untuk memandu LED luaran fizikal kita.

3.3 Amali 3A: LED Terbina Dalam (*Built-In* LED Blink)

Hampir semua papan pembangunan ESP32 didatangkan dengan satu lampu LED kecil yang dipasang siap-siap di atas papan litar (*onboard LED*). LED terbina dalam ini disambungkan terus ke **GPIO 2** pada cip ESP32.Amali ini bertujuan untuk menguji sama ada perkakasan ESP32

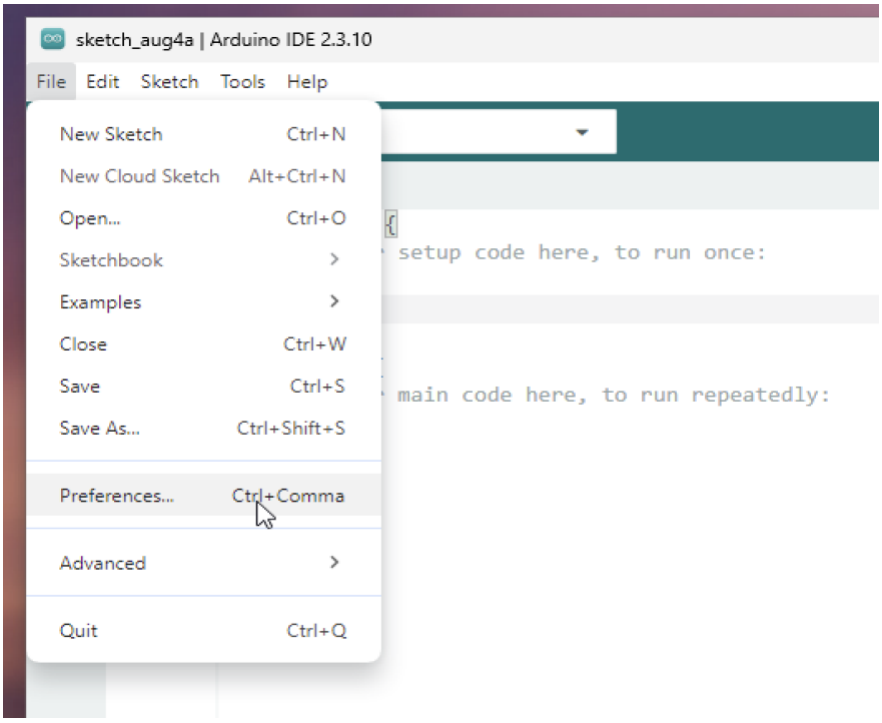
anda berfungsi dengan baik tanpa perlu membuat sebarang pendawaian litar luaran.

3.3.1 Konfigurasi Board Manager ESP32 di Arduino IDE

Sebelum menulis kod, pastikan anda menukar tetapan papan di Arduino IDE:

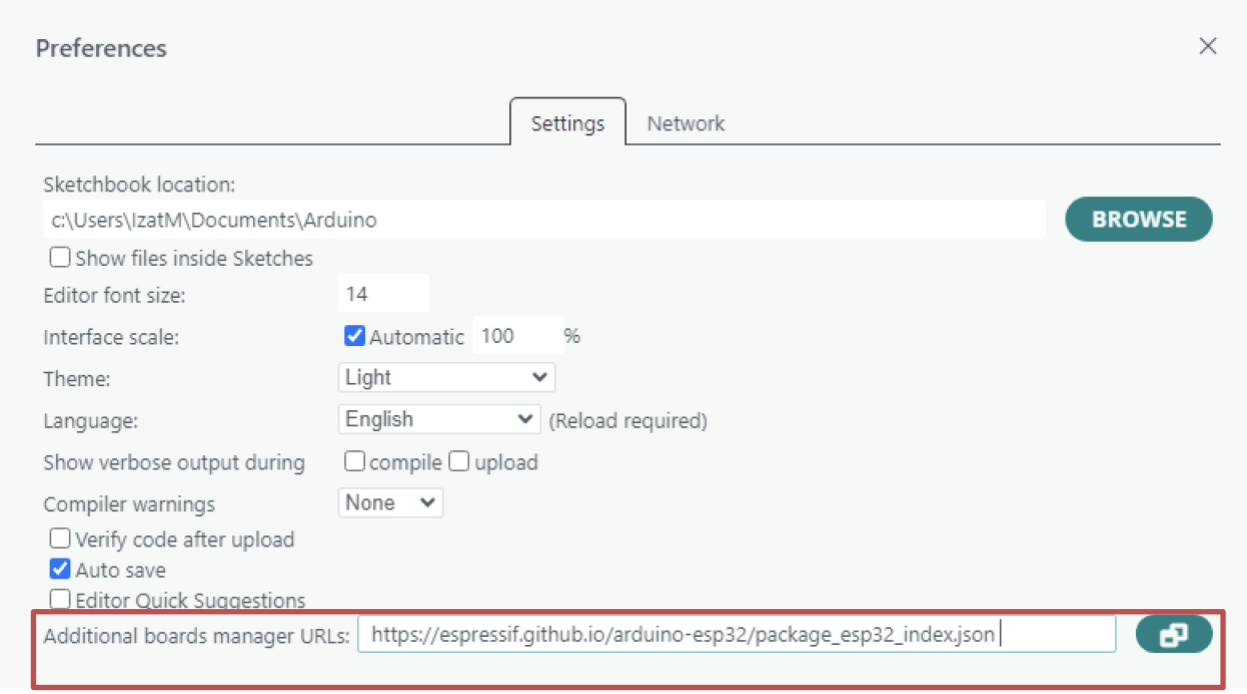
a. Pergi ke **Tools > Board > ESP32 Arduino**.
b. Pilih jenis papan yang sesuai, biasanya **"DOIT ESP32 DEVKIT V1"** atau **"ESP32 Dev Module"**.
c. Pergi ke **Tools > Port** dan pilih Port COM yang aktif.

3.3.2 Kod Program Amali 3A: Pengujian operasi ESP32

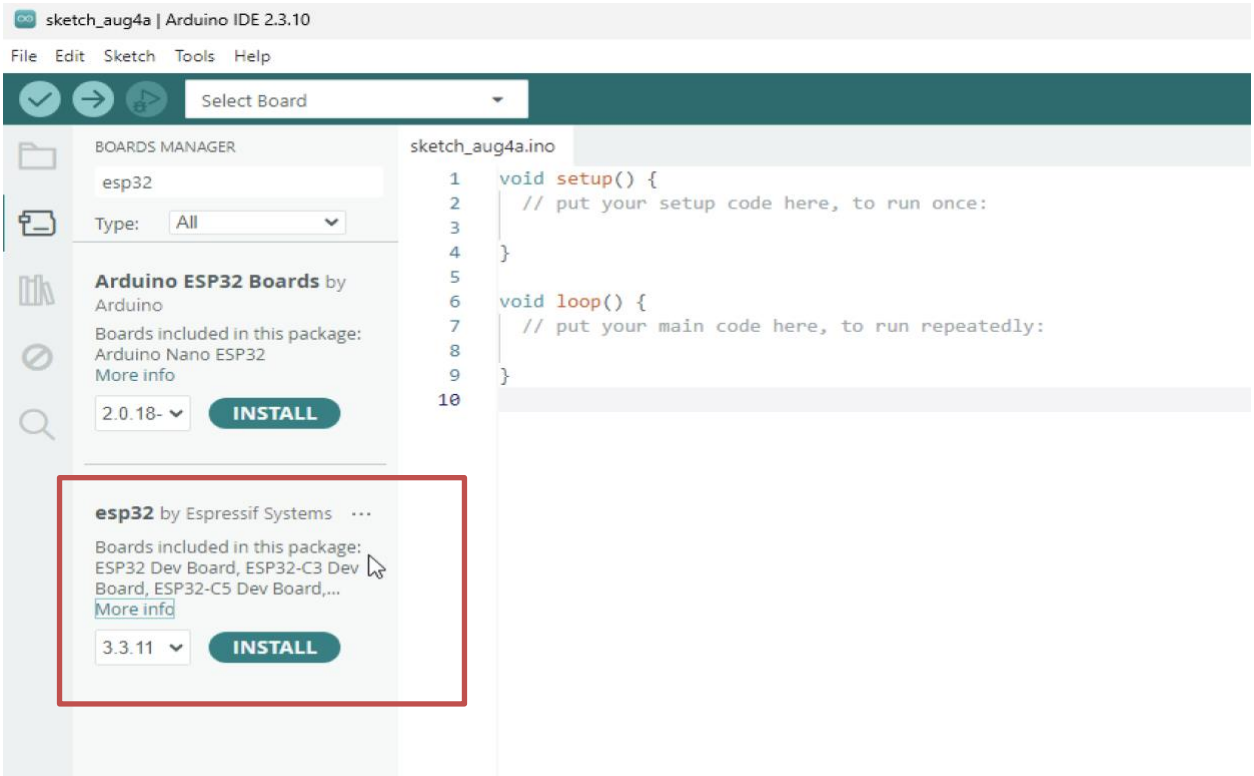



```
https://espressif.github.io/arduino-esp32/package_esp32_index.json
```

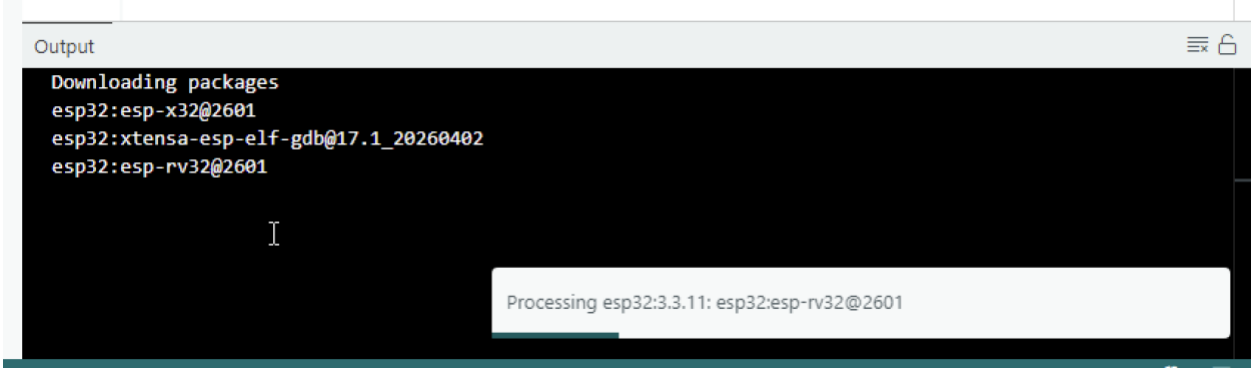
Rajah 3.2 : mengisi indeks JSON untuk memuat turun semua fail sokongan ESP32



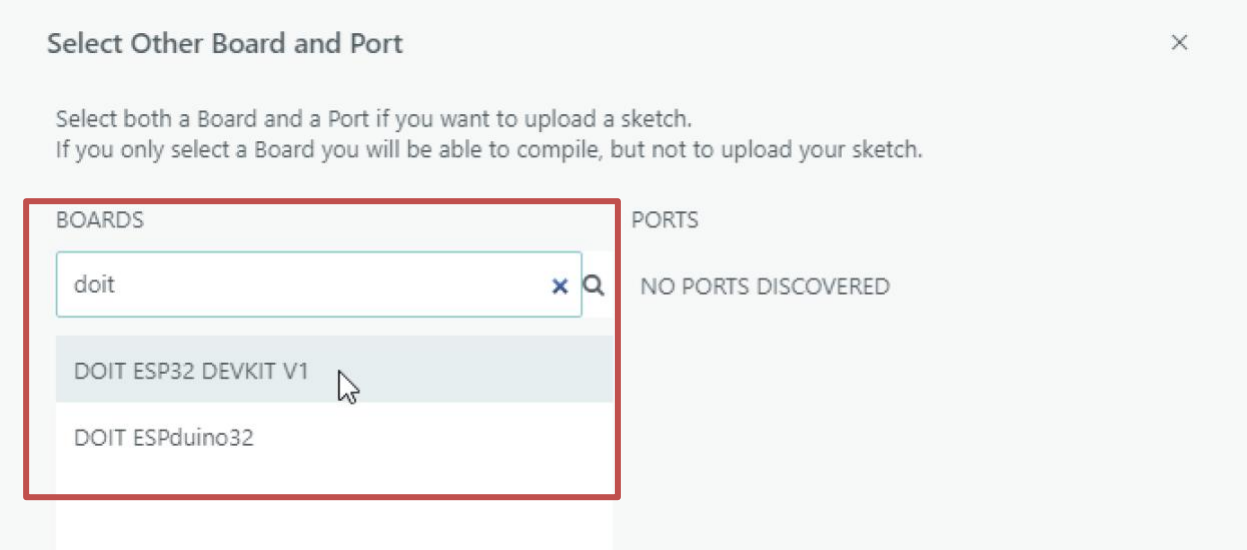
Rajah 3.3: memuatkan URL berkaitan JSON untuk ESP32 dalam bar *preferences*



Rajah 3.4 : memuatkan papan pengawal berkaitan ESP32 untuk sokongan pelaksanaan projek berkaitan



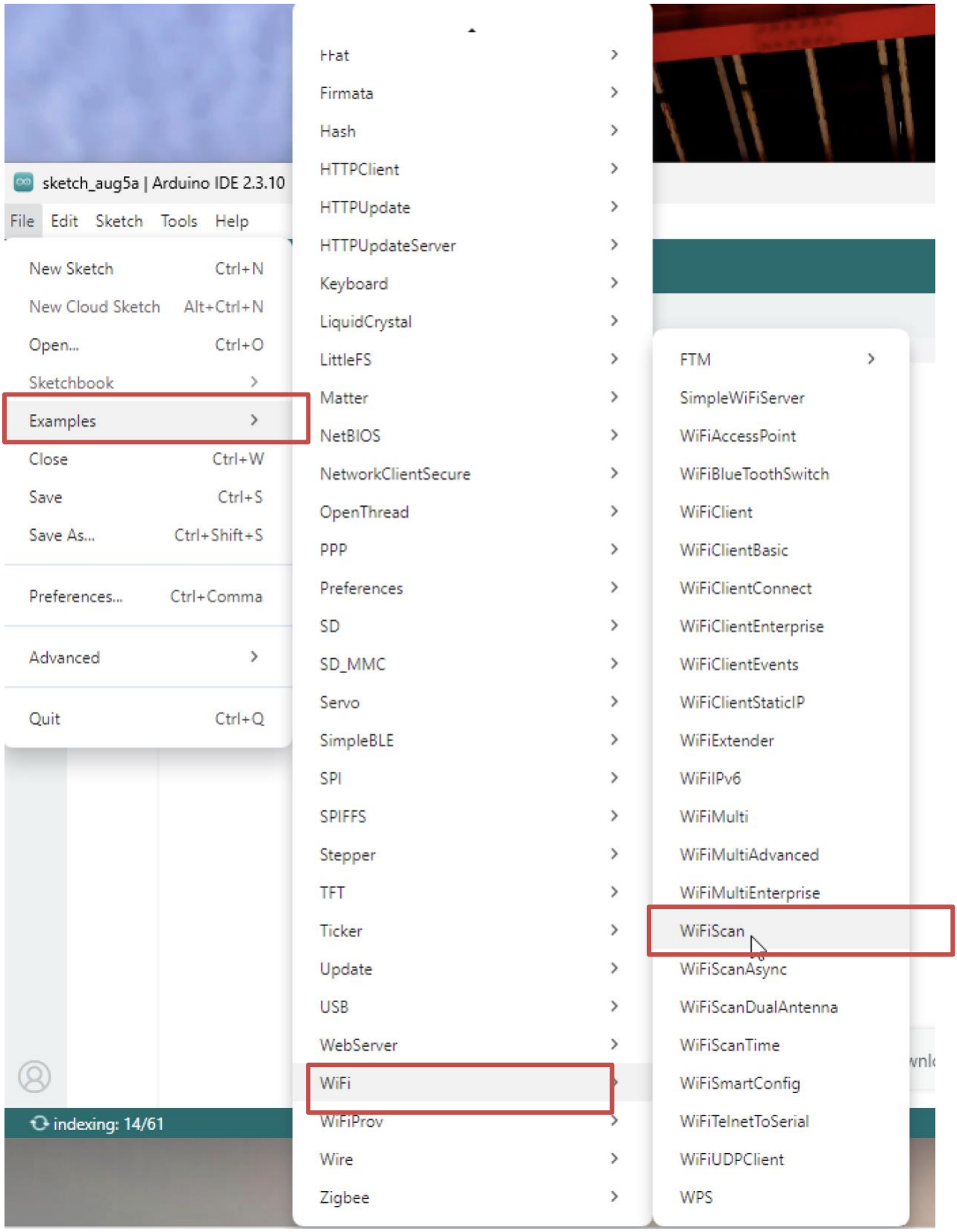
Rajah 3.5 : Menunjukkan pakej berkaitan ESP32 sedang di muat turun.



Rajah 3.6 : Menunjukkan pilihan papan berkaitan ESP32 dimuatkan menunjukkan kaedah berjaya.

Kaedah seterusnya sangat penting kerana kita perlukan aturcara yang telah siap dituliskan oleh pembangun untuk menguji ESP32. Antara fungsi terpenting sudah tentu pengujian *WiFi*. Kerana tanpa fungsi terpenting ini, tak mungkin projek berkaitan IoT boleh dilaksanakan. Anda hanya perlu menekan tab *File* dan pilih *Examples.* Cari opsyen *Wifi* di bawah tab tersebut dan pilih fungsi yang asas iaitu *WifiScan.*

Aturcara yang telah disiapkan ini akan menguji cip *WiFi* yang akan mengesan isyarat *WiFi* yang ada dan dipaparkan untuk kegunaan pengguna. Sila lihat rajah seterusnya untuk pemahaman lanjut.



Rajah 3.7 : Kaedah menggunakan aturcara terbina iaitu *WifiScan*

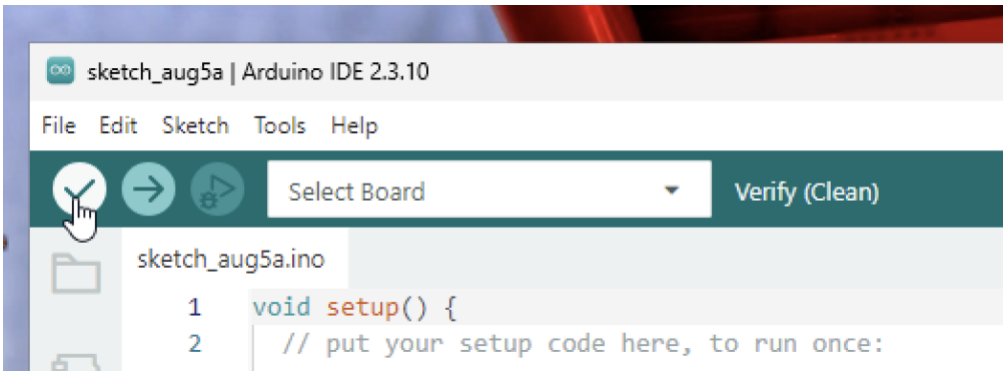
```
/*
 *  This sketch demonstrates how to scan WiFi networks. For chips that
support 5GHz band, separate scans are done for all bands.
 *  The API is based on the Arduino WiFi Shield library, but has
significant changes as newer WiFi functions are supported.
 *  E.g. the return value of `encryptionType()` different because more
modern encryption is supported.
 */
#include <Arduino.h>
#include "WiFi.h"

void setup() {
  Serial.begin(115200);
  // Enable Station Interface
  WiFi.STA.begin();
  Serial.println("Setup done");
}

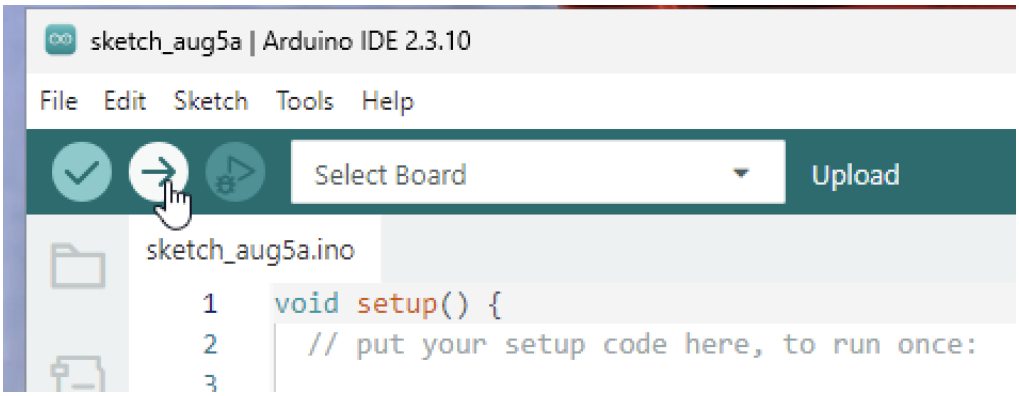
void ScanWiFi() {
  Serial.println("Scan start");
  // WiFi.scanNetworks will return the number of networks found.
  int n = WiFi.scanNetworks();
  Serial.println("Scan done");
  if (n == 0) {
    Serial.println("no networks found");
  } else {
    Serial.print(n);
    Serial.println(" networks found");
    Serial.println("Nr | SSID                                | RSSI | CH
| Encryption");
    for (int i = 0; i < n; ++i) {
      // Print SSID and RSSI for each network found
      Serial.printf("%2d", i + 1);
      Serial.print(" | ");
      Serial.printf("%-32.32s", WiFi.SSID(i).c_str());
      Serial.print(" | ");
      Serial.printf("%4" PRIi32, WiFi.RSSI(i));
      Serial.print(" | ");
      Serial.printf("%2" PRIi32, WiFi.channel(i));
      Serial.print(" | ");
      switch (WiFi.encryptionType(i)) {
        case WIFI_AUTH_OPEN:            Serial.print("open"); break;
        case WIFI_AUTH_WEP:             Serial.print("WEP"); break;
        case WIFI_AUTH_WPA_PSK:         Serial.print("WPA"); break;
        case WIFI_AUTH_WPA2_PSK:        Serial.print("WPA2"); break;
        case WIFI_AUTH_WPA_WPA2_PSK:    Serial.print("WPA+WPA2");
break;
        case WIFI_AUTH_WPA2_ENTERPRISE: Serial.print("WPA2-EAP");
break;
        case WIFI_AUTH_WPA3_PSK:        Serial.print("WPA3"); break;
        case WIFI_AUTH_WPA2_WPA3_PSK:   Serial.print("WPA2+WPA3");
```

```
break;
        case WIFI_AUTH_WAPI_PSK:        Serial.print("WAPI"); break;
        default:                        Serial.print("unknown");
      }
      Serial.println();
      delay(10);
    }
  }

  // Delete the scan result to free memory for code below.
  WiFi.scanDelete();
  Serial.println("-------------------------------------");
}
void loop() {
  Serial.println("-------------------------------------");
  Serial.println("Default wifi band mode scan:");
  Serial.println("-------------------------------------");
#if ESP_IDF_VERSION >= ESP_IDF_VERSION_VAL(5, 4, 2)
  WiFi.setBandMode(WIFI_BAND_MODE_AUTO);
#endif
  ScanWiFi();
#if CONFIG_SOC_WIFI_SUPPORT_5G
  // Wait a bit before scanning again.
  delay(1000);
  Serial.println("-------------------------------------");
  Serial.println("2.4 Ghz wifi band mode scan:");
  Serial.println("-------------------------------------");
  WiFi.setBandMode(WIFI_BAND_MODE_2G_ONLY);
  ScanWiFi();
  // Wait a bit before scanning again.
  delay(1000);
  Serial.println("-------------------------------------");
  Serial.println("5 Ghz wifi band mode scan:");
  Serial.println("-------------------------------------");
  WiFi.setBandMode(WIFI_BAND_MODE_5G_ONLY);
  ScanWiFi();
#endif
  // Wait a bit before scanning again.
  delay(10000);
}
```

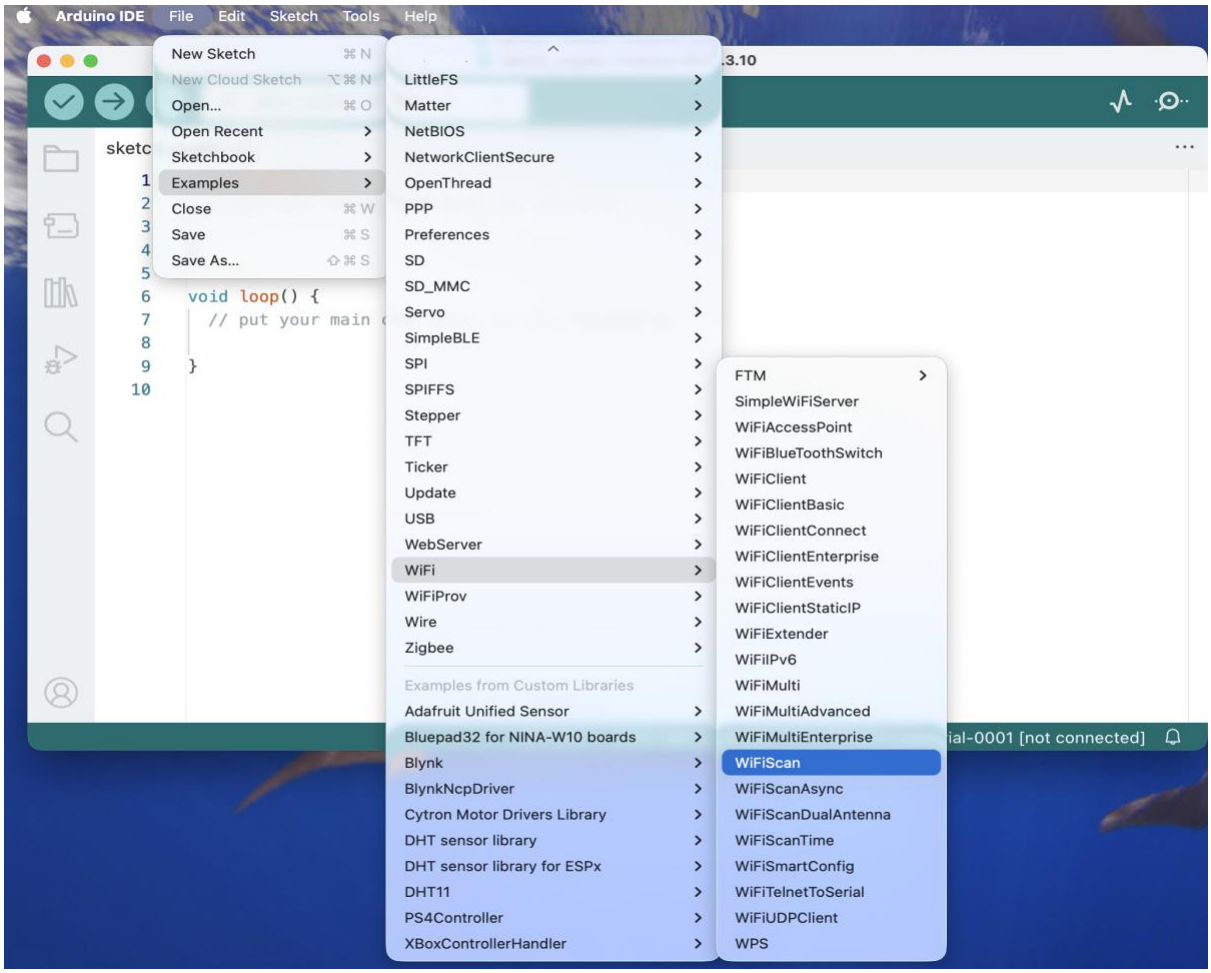


Rajah 3.8 : Untuk *verify* perlu guna klik pada button yang berikut



Rajah 3.9 : Untuk *upload* perlu guna klik pada button yang berikut

SOP Memuat Naik ESP32: Jika anda melihat perkataan **"Connecting..."** berserta simbol titik-titik (___..___) di bahagian panel bawah Arduino IDE semasa memuat naik sehingga muncul **Done Uploading.**



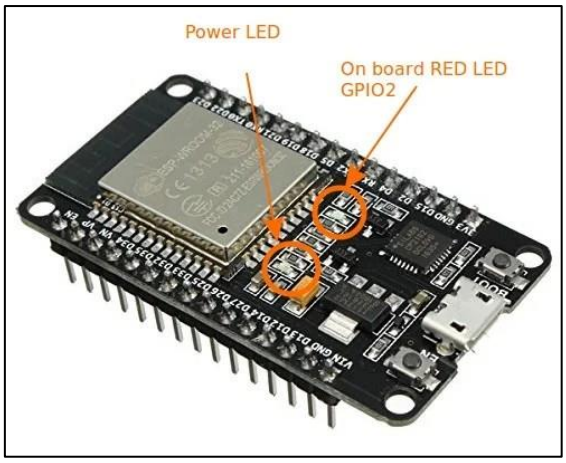
Rajah 3.10 : Antaramuka ArduinoIDE untuk pengguna MacOS

Salah satu kaedah lain untuk ujian ESP32 adalah dengan melaksanakan pengujian built in LED yang bersambung dengan GPIO2. Jika LED beroperasi mengikut program di bawah, maka kita tahu bahawa modul tersebut berfungsi dengan baik.

```
// Menetapkan nama pembolehubah 'ledPin' yang merujuk kepada GPIO 2 ESP32
const int ledPin = 2; // LED terbina dalam ESP32 disambung ke GPIO 2
void setup()    // Mengkonfigurasikan pin GPIO 2 sebagai OUTPUT digital
  pinMode(ledPin, OUTPUT);
}

void loop() {

  digitalWrite(ledPin, HIGH); // 1. Menghantar isyarat HIGH (voltan 3.3V) untuk menghidupkan LED
  delay(1000);                // Tunggu selama 1 saat (1000 milisaat)
                              // 2. Menghantar isyarat LOW (voltan 0V) untuk mematikan LED
  digitalWrite(ledPin, LOW);
  delay(1000);                // Tunggu selama 1 saat
}
```



Rajah 3.11: Kedudukan LED yang terbina dalam ESP32

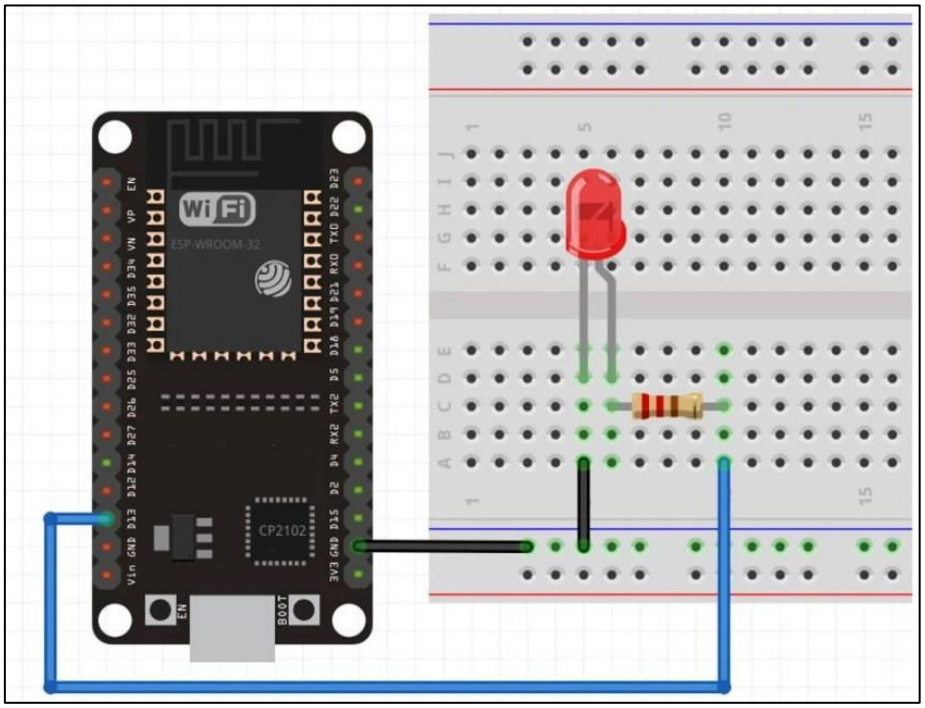
3.4 Amali 3B: Mengawal LED Luaran Menggunakan BlynkIoT (Pin Maya)

Dalam latihan bersepadu ini, kita akan menyambungkan sebuah LED luaran fizikal pada **GPIO 5** ESP32, dan mengawal status buka/tutupnya secara wayarles melalui internet menggunakan butang digital yang dibina di dalam aplikasi Blynk(Pin Maya V0).

3.4.1 Perkakasan yang Diperlukan

1. Papan Pembangunan ESP32.
2. 1 unit LED 5mm.
3. 1 unit Perintang 220 Ohm atau 330 Ohm
4. Papan reka bentuk (*Breadboard*) dan Wayar jumper.

3.4.2 Rajah Sambungan Litar



3.4.3 Kod Program Amali 3B: BlynkBlink LED Luaran (ESP32 WiFi)

Kod di bawah diambil berdasarkan contoh rasmi *Getting Started - BlynkBlink* daripada BlynkExample Browser. Kita menggunakan fungsi makro BLYNK_WRITE(V0) untuk mendengar arahan perubahan suis butang digital daripada pelayan awan Blynk.

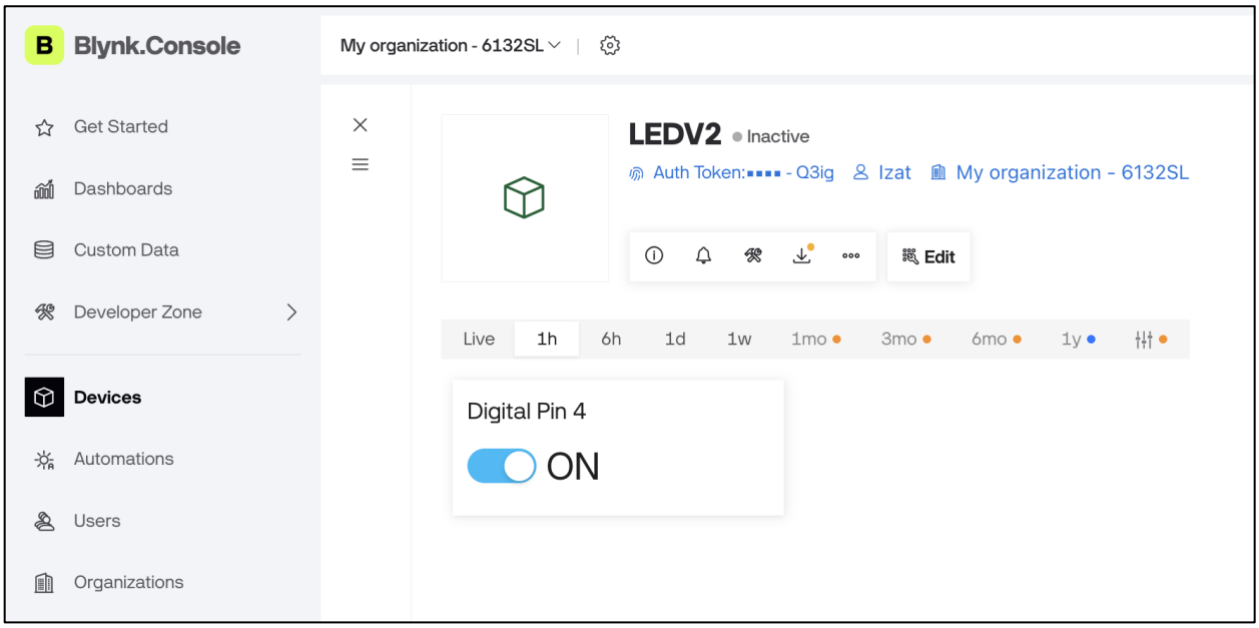


Rajah 3.12 : Antaramuka *Blynk Example*

3.4.4 Panduan Penyediaan *Widget* Suis pada Aplikasi *Blynk*

Untuk membolehkan kod di atas berfungsi dengan sempurna, pelajar perlu membuat tetapan berikut pada aplikasi *Blynk*di telefon pintar atau *BlynkWeb Console*:

1. **Cipta Datastream DigitalPin:**
   - Pilih jenis ***DigitalPin*** . Setkan pilihan Pin kepada **D4**.
   - Pilih jenis data sebagai **Integer** (Nilai minimum 0, nilai maksimum 1).
2. **Bina *Widget* Kawalan (Button):**
   - Seret *widget* ***Button / Switch*** ke atas kanvas *dashboard* anda.
   - Pergi ke tetapan widget tersebut dan hubungkannya ke *Datastream* **D4 (LED Control)** yang telah dibina tadi.
   - Setkan mod butang tersebut kepada **"*Switch*"** (bukan *Push*) agar ia mengekalkan keadaan ON/OFF setiap kali ditekan.
3. **Ujilari Projek:**
   - Tekan butang **Play** di bahagian atas aplikasi Blynk.
   - Toggle suis V0 tersebut. LED luaran pada GPIO 5 ESP32 anda akan menyala dan terpadam secara masa nyata mengikut status butang!



Rajah 3.13 : Binaan butang dengan menggunakan aplikasi *Blynk*

Panduan yang lengkap ada dalam video di bawah

https://www.youtube.com/watch?v=-icFb4rgnPA&t=6s

Nota : SILA IKUTI PANDUAN YANG LEBIH LENGKAP DALAM EBOOK IOT SIRI 2 AKAN DATANG

Rujukan

1. Real-Time Industrial IoT Analytics | InfluxData, accessed August 27, 2025, https://www.influxdata.com/solutions/industrial-iot/
2. What is MQTT? - MQTT Protocol Explained - AWS, accessed August 27, 2025, https://aws.amazon.com/what-is/mqtt/
3. LAPORAN PROJEK SMART STREET LIGHT BASED IOT - Politeknik ..., accessed August 27, 2025, http://repository.psa.edu.my/bitstream/123456789/4333/1/SMART%20STREET%20LIGHT%20BASED%20IOT%2008DKA20F1003%20SITI%20AISYAH%20BINTI%20SALIM.pdf
4. Getting Started with Arduino 2. August 01 2026 https://store.arduino.cc/products/arduino-uno-rev3
5. Getting Started with Blynk. August 02 2026 https://www.blynk.io/getting-started