BENGKEL · SEKOLAH RENDAH

Rumah Pintar Oyen

Belajar Arduino dan ESP32, minta Gemini tulis kod, dan lihat litar kita hidup dalam simulator.

Tahun 4 hingga 6 · Arduino Uno & ESP32 · Gemini · Makmal Arduino Maya



Kandungan

Pengenalan 4

Perjalanan kita hari ini 4

Bab 1 · Otak kecil 6

Apakah Arduino? 6

Tiada papan sebenar? Tiada masalah! 6

Bab 2 · Pintu-pintu pin 8

Pin ialah pintu 8

Masuk atau keluar? 9

Aktiviti: Isih komponen 10

Bab 3 · Sambung dengan betul 11

Mengenali LED 11

Litar LED yang betul 11

Butang: cara paling mudah 12

Buzzer dan sensor 13

Pin mana yang boleh dipakai? 13

Bab 4 · Pelan Pin 15

Pelan Pin: peta litar kita 15

Aktiviti: Buat Pelan Pin kamu 16

Bab 5 · Bercakap dengan Gemini 17

Gemini: kawan yang pandai menulis kod 17

Lima bahan prompt yang hebat 17

Aktiviti: Bahan apa yang hilang? 18

Menulis TUGAS yang jelas 18

Aktiviti: Pembina Arahan 20

Semak: adakah kod ikut Pelan Pin? 20

Bab 6 · Bina & Jalankan 22

Membuka Makmal Arduino Maya 22

Kali pertama: lesen Wokwi 22

Tampal, bina, jalankan 23

Melukis litar di wokwi.com 23

Alamak, ada ralat! 24

Aktiviti: Fahami mesej ralat 24

Bab 7 · Projek Rumah Pintar 26

Projek Rumah Pintar Oyen 26

Tahap 3: “Oyen dah balik!” melalui Wi-Fi 26

Senarai semak projek 27

Penutup 28

Tahniah, jurutera cilik! 28

Lampiran A · Nota cikgu **29**

Lampiran B · Jawapan aktiviti **32**

Lampiran C · Glosari **33**

Pengenalan

Ini **Mia**. Mia ada seekor kucing oren bernama **Oyen** . Oyen tidur dalam sebuah rumah kotak kecil. Mia mahu jadikan rumah itu **rumah pintar**: ada lampu, ada loceng, dan boleh beritahu Mia apabila Oyen sudah pulang.

Mia tidak pandai menulis kod. Tidak mengapa! Mia akan belajar **tiga perkara** sahaja:

Sambung
Cara menyambung lampu, butang dan bunyi ke papan Arduino.

Minta
Cara meminta Gemini menulis kod dengan arahan yang jelas.

Jalankan
Cara menjalankan kod dalam simulator dan membetulkannya.

Perjalanan kita hari ini

Setiap bahagian menjawab soalan yang timbul daripada bahagian sebelumnya.

Bahagian	Soalan	Masa
1 Otak kecil	Apakah Arduino dan ESP32?	25 minit
2 Pintu-pintu pin	Bagaimana otak itu bercakap dengan lampu dan butang?	30 minit
3 Sambung dengan betul	Bagaimana menyambung LED, butang dan buzzer tanpa merosakkannya?	40 minit
4 Pelan Pin	Bagaimana mencatat apa disambung ke mana?	20 minit
5 Bercakap dengan Gemini	Bagaimana meminta kod yang betul?	45 minit
6 Bina & Jalankan	Bagaimana menjalankan kod dan membetulkan ralat?	40 minit
7 Projek Rumah Pintar	Boleh kita bina rumah pintar Oyen sepenuhnya?	60 minit+

Masa hanya cadangan. Cikgu boleh pecahkan kepada beberapa sesi.

BAB 1

Otak kecil

Apakah Arduino?

> Mia mahu lampu rumah Oyen menyala sendiri. Siapa yang akan memberi arahan kepada lampu itu? Jawapannya: sebuah **otak kecil**.



Arduino ialah papan kecil yang ada **mikropengawal**. Ia seperti otak kecil. Kita beri ia satu senarai arahan (**kod**), dan ia akan ikut arahan itu berulang-ulang tanpa penat.

Arduino Uno

Papan paling popular untuk belajar. Ada 14 pin digital. Sesuai untuk lampu, butang, bunyi dan sensor.

Tiada Wi-Fi.

ESP32

Lebih laju dan ada **Wi-Fi** serta Bluetooth. Boleh hantar mesej ke Internet.

Kita guna ESP32 untuk projek yang perlu Internet.

Fakta menarik: mesin basuh, ketuhar gelombang mikro dan alat kawalan jauh TV di rumah kamu juga ada mikropengawal di dalamnya!

Tiada papan sebenar? Tiada masalah!

Hari ini kita guna **simulator**. Simulator meniru papan Arduino di skrin komputer. Lampu akan menyala, butang boleh ditekan, dan bunyi boleh didengar, semuanya di skrin.

- ✔ Tiada wayar yang tercabut.
- ✔ Tiada komponen yang terbakar jika tersalah sambung.
- ✔ Kod yang sama boleh dimuat naik ke papan sebenar nanti.

Aplikasi kita: Makmal Arduino Maya. Cikgu sudah memasangnya. Cari ikon **Makmal ESP32** atau **Makmal Uno** di Desktop.

✎ AKTIVITI 1 · BETUL ATAU SALAH?

1. Arduino boleh berfikir sendiri tanpa kod.

- □ A. Betul
- □ B. Salah

2. ESP32 boleh bersambung ke Wi-Fi.

- □ A. Betul
- □ B. Salah

SOALAN SETERUSNYA

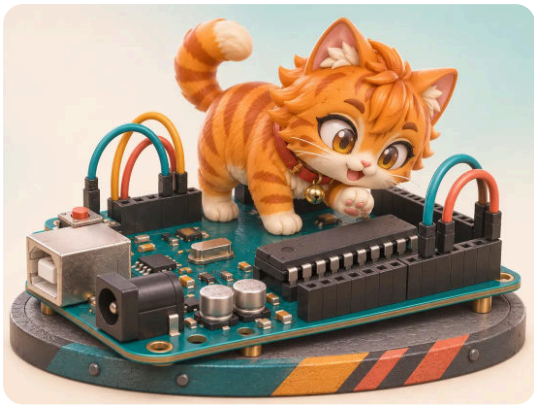
Otak kecil sudah ada. Tetapi bagaimana otak itu **bercakap** dengan lampu dan butang?

BAB 2

Pintu-pintu pin

Pin ialah pintu

> Bayangkan otak kecil itu duduk dalam sebuah bilik yang banyak **pintu**. Melalui setiap pintu, ia boleh menghantar arahan keluar atau menerima berita masuk.



Pintu-pintu itu dipanggil **pin GPIO**. Setiap pin ada **nombor**. Kod akan menyebut nombor pin, contohnya pin 13 atau GPIO 26.



Pin kuning ialah pintu bernombor. Pin kelabu ialah kuasa: **GND** (negatif, “bumi”) dan **5V** atau **3V3** (positif).

Masuk atau keluar?

OUTPUT Keluar

Otak **menghantar** arahan keluar. Contoh: nyalakan LED, bunyikan buzzer, pusingkan motor.

Dalam kod: `pinMode(13, OUTPUT);`

INPUT Masuk

Otak **menerima** berita dari luar. Contoh: butang ditekan, suhu bilik, cahaya matahari.

Dalam kod: `pinMode(4, INPUT_PULLUP);`



Setiap pin hanya boleh jadi **satu** pada satu masa: input *atau* output. Sebab itu kita perlu tahu setiap komponen jenis yang mana.

Digital bermaksud hanya dua keadaan: **HIGH** (hidup, 1) atau **LOW** (mati, 0). Seperti suis lampu di rumah: buka atau tutup.

Aktiviti: Isih komponen

✎ AKTIVITI 2 · INPUT ATAU OUTPUT?

Tandakan (✓) jenis setiap komponen.

Komponen	Input	Output
LED	☐	☐
Butang	☐	☐
Buzzer	☐	☐
Sensor suhu	☐	☐
Potensiometer	☐	☐
Motor servo	☐	☐
Sensor cahaya	☐	☐
Skrin LCD	☐	☐

SOALAN SETERUSNYA

Kita tahu LED ialah output. Tetapi bolehkah LED disambung terus ke pin begitu sahaja?

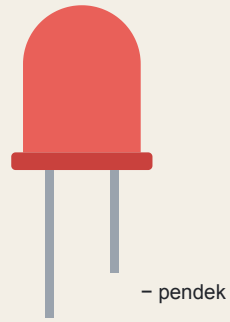
BAB 3

Sambung dengan betul

Mengenali LED

Mia sambung LED terus ke pin. Dalam dunia sebenar, LED itu boleh **terbakar**! Kenapa?





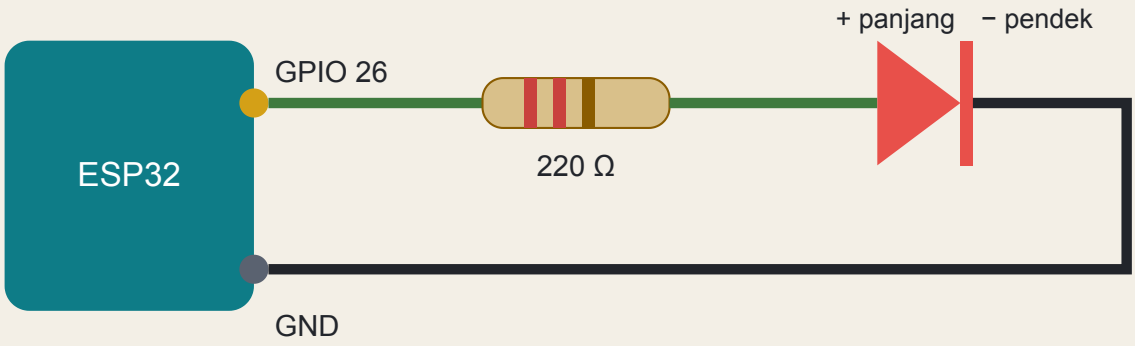
Kaki **panjang** = positif (**anod**). Kaki **pendek** = negatif (**katod**).

LED ialah lampu kecil. Arus elektrik hanya boleh lalu **satu arah**: masuk ikut kaki panjang, keluar ikut kaki pendek.

Jika terbalik, LED tidak akan menyala.

LED juga tidak tahan arus yang terlalu kuat. Kita perlu "pengawal pintu" untuk mengehadkan arus. Pengawal itu dipanggil **perintang**.

Litar LED yang betul



Pin → perintang 220 Ω → kaki panjang LED. Kaki pendek → GND. Litar mesti membuat **bulatan penuh**.

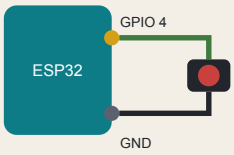
1. Wayar dari **pin** (contoh GPIO 26) ke satu hujung **perintang**.
2. Hujung lain perintang ke **kaki panjang** LED.
3. **Kaki pendek** LED ke **GND**.

Perintang boleh diletak di sebelah kaki pendek juga. Yang penting, ia ada dalam bulatan itu.

Butang: cara paling mudah

> Mia mahu Oyen boleh “tekan loceng”. Kita perlukan sebuah **butang**.





Satu kaki butang ke pin, satu lagi ke GND.

Sambung butang antara **pin** dan **GND**. Tidak perlu perintang!

Dalam kod, kita guna `INPUT_PULLUP`. Ia menghidupkan perintang kecil yang tersembunyi di dalam papan.

> **Ingat, ia terbalik:**
> Butang **tidak ditekan** → pin baca **HIGH**
> Butang **ditekan** → pin baca **LOW**

Beritahu Gemini perkara ini supaya kodnya betul (kita belajar di Bahagian 5).

Buzzer dan sensor

Buzzer (output)

Mengeluarkan bunyi. Kaki **+** ke pin, kaki **–** ke GND.

Dalam kod, bunyi dibuat dengan `tone(pin, frekuensi)`. Frekuensi tinggi = bunyi nyaring.

Sensor suhu DHT22 (input)

Tiga kaki: **VCC** ke 3V3, **GND** ke GND, **SDA** (data) ke satu pin, contoh GPIO 15.

Perlukan **pustaka** DHT. Makmal Arduino Maya sudah menyediakannya.

Potensiometer (input)

Tombol pusing. Kaki tengah ke pin **analog** (Uno: A0; ESP32: GPIO 34). Kaki tepi ke 3V3/5V dan GND.

Analog bermaksud nilai banyak, bukan hidup/mati sahaja: 0 hingga 1023 (Uno) atau 0 hingga 4095 (ESP32).

Sensor cahaya LDR (input)

Mengesan terang atau gelap. Dalam Wokwi, modul LDR ada kaki **AO** (analog) ke pin analog.

Sesuai untuk lampu malam automatik.

Pin mana yang boleh dipakai?

Bukan semua pin sesuai untuk murid. Sesetengah pin ada tugas rahsia ketika papan dihidupkan. Guna senarai ini:

Papan	Selamat untuk LED, butang, buzzer	Analog (sensor)	Elakkan
ESP32	4, 5, 13, 14, 16, 17, 18, 19, 21, 22, 23, 25, 26, 27, 32, 33	32, 33, 34, 35 (34 & 35 input sahaja)	0, 1, 2, 3, 6 hingga 12, 15*
Uno	2 hingga 13	A0 hingga A5	0 dan 1 (untuk Serial)

* GPIO 15 masih boleh untuk sensor DHT22 seperti dalam contoh Wokwi, tetapi elakkan untuk butang.

✎ AKTIVITI 3 · UJI SAMBUNGAN

3. LED Mia tidak menyala. Kaki panjang disambung ke GND dan kaki pendek ke perintang. Apa masalahnya?

- □ A. Perintang terlalu kecil
- □ B. Pin salah nombor
- □ C. LED terbalik
- □ D. Buzzer rosak

4. Butang disambung ke GPIO 4 dan GND dengan INPUT_PULLUP. Apakah bacaan pin apabila butang ditekan?

- □ A. HIGH
- □ B. LOW
- □ C. 1023
- □ D. Tiada bacaan

5. Pin manakah paling sesuai untuk LED pada ESP32?

- □ A. GPIO 6
- □ B. GPIO 0
- □ C. GPIO 34
- □ D. GPIO 26

SOALAN SETERUSNYA

Rumah Oyen akan ada banyak komponen. Bagaimana kita ingat apa disambung ke pin mana?

BAB 4

Pelan Pin

Pelan Pin: peta litar kita

> Mia hampir terlupa LED merah di pin berapa. Mulai sekarang, Mia akan tulis **Pelan Pin** dahulu sebelum menyambung apa-apa.



Pelan Pin ialah jadual kecil: komponen apa, disambung ke pin mana, dan jenisnya. Pelan ini sangat penting kerana:

- ✔ Kita sambung wayar ikut pelan.
- ✔ Kita beri pelan yang **sama** kepada Gemini.
- ✔ Jadi kod Gemini **sepadan** dengan litar kita.

Komponen	Pin	Jenis	Nota
LED merah	GPIO 26	OUTPUT	melalui perintang 220 Ω
Butang loceng	GPIO 4	INPUT	ke GND, INPUT_PULLUP
Buzzer	GPIO 25	OUTPUT	kaki – ke GND

Aktiviti: Buat Pelan Pin kamu

✎ AKTIVITI 4 · PELAN PIN RUMAH OYEN

Isi Pelan Pin untuk rumah pintar Oyen. Guna jadual pin selamat di halaman sebelumnya. Pastikan tiada dua komponen berkongsi pin yang sama.

Papan: □ ESP32 □ Arduino Uno

Komponen	Pin	Jenis	Nota sambungan
LED merah		□ IN □ OUT	
Butang		□ IN □ OUT	
Buzzer		□ IN □ OUT	
		□ IN □ OUT	
		□ IN □ OUT	

SOALAN SETERUSNYA

Pelan sudah siap. Sekarang, bagaimana kita meminta Gemini menulis kod yang ikut pelan ini?

BAB 5

Bercakap dengan Gemini

Gemini: kawan yang pandai menulis kod

> Mia menaip kepada Gemini: *“buat kod lampu”*. Gemini memberi kod… untuk papan lain, pin lain, dan lampu yang berkelip laju. Litar Mia tidak menyala langsung!



Gemini sangat pandai, tetapi ia **tidak dapat melihat** litar kita. Ia hanya tahu apa yang kita tulis. Arahan yang kita beri kepada Gemini dipanggil **prompt**.

Peraturan emas: Jika Gemini terpaksa **meneka**, ia mungkin salah teka. Jadi, jangan biar Gemini meneka!

Jaga keselamatan: jangan taip nama penuh, alamat, nombor telefon atau kata laluan dalam Gemini. Gunakan akaun sekolah (DELIMa) yang cikgu berikan.

Lima bahan prompt yang hebat

Seperti memasak nasi lemak, prompt yang sedap perlukan bahan yang cukup:

1 · PAPAN
Papan apa? “ESP32 DevKit” atau “Arduino Uno”.

2 · PELAN PIN
Setiap komponen dan pinnya.

3 · TUGAS
Apa yang berlaku, langkah demi langkah.

4 · PERATURAN
Kod Arduino penuh, satu blok, jangan ubah pin.

5 · SEMAKAN
Minta mesej Serial supaya kita tahu ia berjalan.

Contoh prompt yang ada kelima-lima bahan:

> Tulis kod Arduino untuk papan ESP32 DevKit (ESP32 Arduino core 2.0.17).
> Sambungan: LED merah di GPIO 26 melalui perintang 220 ohm. Butang di GPIO 4 ke GND, guna INPUT_PULLUP (tekan = LOW).
> Tugas: Apabila butang ditekan, LED menyala selama 3 saat, kemudian padam.
> Peraturan: Beri kod penuh dalam satu blok. Jangan ubah nombor pin. Tulis komen ringkas dalam Bahasa Melayu.
> Semakan: Cetak "Butang ditekan!" ke Serial Monitor pada 115200 baud setiap kali butang ditekan.

Aktiviti: Bahan apa yang hilang?

✎ AKTIVITI 5 · DETEKTIF PROMPT

6. “Tulis kod Arduino untuk ESP32. LED di GPIO 26. Beri kod penuh dalam satu blok.”

- ☐ A. Papan
- ☐ B. Pelan Pin
- ☐ C. Tugas
- ☐ D. Peraturan

7. “Tulis kod Arduino untuk Arduino Uno. Apabila butang ditekan, buzzer berbunyi. Beri kod penuh.”

- ☐ A. Papan
- ☐ B. Pelan Pin
- ☐ C. Tugas
- ☐ D. Semakan

8. “LED di pin 13 berkelip setiap 1 saat. Beri kod penuh, jangan ubah pin.”

- ☐ A. Papan
- ☐ B. Pelan Pin
- ☐ C. Tugas
- ☐ D. Peraturan

Menulis TUGAS yang jelas

Bahan ketiga, **TUGAS**, paling susah. Tulis seperti resipi: **langkah demi langkah**, dengan **nombor** dan **masa**.

KABUR

“Buat lampu isyarat.”

JELAS

“Ulang selama-lamanya: LED hijau menyala 5 saat, kemudian LED kuning 2 saat, kemudian LED merah 5 saat. Hanya satu LED menyala pada satu masa.”

Perkataan yang membantu:

Untuk	Guna perkataan
Masa	“selama 3 saat”, “setiap 500 milisaat”
Syarat	“Apabila butang ditekan…”, “Jika suhu lebih daripada 30 °C…”
Ulangan	“Ulang selama-lamanya”, “sebanyak 3 kali”
Urutan	“Mula-mula… kemudian… akhir sekali…”

✎ AKTIVITI 6 · PILIH TUGAS YANG PALING JELAS

9. Mia mahu loceng pintu Oyen. Tugas mana paling jelas?

□ A. Buat loceng.

□ B. Bila tekan butang, bunyi.

□ C. Apabila butang ditekan, buzzer berbunyi 1000 Hz selama 0.5 saat, kemudian senyap. Jika butang ditekan lagi, ulang.

□ D. Buat buzzer bunyi kuat-kuat.

Aktiviti: Pembina Arahan

✎ AKTIVITI 7 · BINA PROMPT KAMU

Lengkapkan templat ini, kemudian taip dalam Gemini.

Tulis kod Arduino untuk papan _____.

Sambungan:
- _____ di pin _____ (_____).
- _____ di pin _____ (_____).
- _____ di pin _____ (_____).

Tugas:

_____

_____

_____

Peraturan: Beri kod penuh dalam satu blok sahaja. Jangan ubah nombor pin. Tulis komen ringkas dalam Bahasa Melayu.

Semakan: Cetak mesej ringkas ke Serial Monitor setiap kali sesuatu berlaku.

Semak: adakah kod ikut Pelan Pin?

Gemini kadang-kadang menukar nombor pin. Sebelum menjalankan kod, cari nombor pin dalam kod dan bandingkan dengan Pelan Pin. Biasanya ia di bahagian atas kod, seperti `const int LED = 26;`.

✎ AKTIVITI 8 · PEMERIKSA PIN

Cari setiap nombor pin dalam kod Gemini (cari `const int`, `#define` dan `pinMode`). Bandingkan dengan Pelan Pin kamu.

Komponen	Pin dalam Pelan Pin	Pin dalam kod Gemini	Sama?
			□ Ya □ Tidak
			□ Ya □ Tidak
			□ Ya □ Tidak

Tip: halaman web modul ini ada **Pemeriksa pin** automatik.

SOALAN SETERUSNYA

Kod sudah ada dan pin sudah sepadan. Bagaimana kita menjalankannya dalam simulator?

BAB 6

Bina & Jalankan

Membuka Makmal Arduino Maya

1. Klik dua kali **Makmal ESP32** (atau **Makmal Uno**) di Desktop.
2. Fail `sketch.ino` terbuka. Inilah tempat kod.
3. Lihat bar biru di bawah. Itu butang-butang kita:



Butang	Fungsi
▶ Bina & Jalankan (F5)	Tukar kod kepada bahasa mesin, kemudian buka simulator.
■ Henti	Hentikan simulator.
Salin untuk Gemini	Salin kod dan mesej ralat, siap untuk ditampal dalam Gemini.
Projek Baharu	Buat folder projek baharu.
Lesen Wokwi	Kali pertama sahaja (dan sebulan sekali): aktifkan simulator.

Kali pertama: lesen Wokwi

Simulator Wokwi percuma, tetapi perlu **lesen** yang diperbaharui lebih kurang sebulan sekali. Cikgu mungkin sudah melakukannya.

1. Tekan **Lesen Wokwi**, pilih **Dapatkan lesen melalui pelayar**.
2. Pelayar web terbuka. Log masuk akaun Wokwi (minta bantuan cikgu).
3. Tekan **GET YOUR LICENSE**.
4. Pelayar bertanya “Buka Visual Studio Code?”. Tekan **Buka**.
5. Kembali ke aplikasi. Mesej “License activated” muncul. Siap!

Jika aplikasi tidak terbuka semula: tekan Lesen Wokwi sekali lagi dan pilih **Masukkan kunci lesen secara manual**. Cikgu akan bantu.

Simulator perlukan Internet. Membina kod (compile) tidak perlu Internet.

Tampal, bina, jalankan



1. Dalam Gemini, tekan butang **salin** pada blok kod.
2. Dalam `sketch.ino`, tekan **Ctrl+A** (pilih semua), kemudian **Ctrl+V** (tampal). Kod lama diganti.
3. Tekan ▶ **Bina & Jalankan** atau **F5**.
4. Tunggu mesej **"Membina kod..."**. Kali pertama mungkin ambil masa 1 minit.
5. Simulator terbuka. Lihat LED, tekan butang dengan tetikus, dengar buzzer!
6. Mesej Serial muncul di bahagian **Terminal** di bawah.

Litar dalam simulator datang daripada fail `diagram.json`. Jika litar kamu berbeza (contoh: ada buzzer), cikgu akan beri `diagram.json` yang betul, atau kamu boleh lukis litar di wokwi.com (halaman seterusnya).

Melukis litar di wokwi.com

Kita boleh **melukis** litar di laman wokwi.com, kemudian bawa lukisan itu ke Makmal Arduino Maya.

1. Buka **wokwi.com**, pilih **ESP32** atau **Arduino Uno**, kemudian projek kosong.
2. Tekan butang **+** (biru) untuk tambah komponen: LED, Resistor, Pushbutton, Buzzer.
3. Klik pada hujung kaki untuk mula wayar, klik pada pin untuk tamatkan. Ikut Pelan Pin!
4. Klik perintang, tukar nilai kepada **220**.
5. Klik tab **diagram.json**, tekan Ctrl+A dan Ctrl+C.
6. Dalam Makmal Arduino Maya, buka `diagram.json`, tekan Ctrl+A dan Ctrl+V. Simpan.

Petua: jangan tekan butang "Play" di wokwi.com. Kita hanya melukis di sana. Kita menjalankan kod di Makmal Arduino Maya, jadi tiada iklan dan tiada had.

Alamak, ada ralat!

Mia tekan Bina & Jalankan. Muncul mesej merah: **“Kod ada ralat.”** Mia tidak panik. Ralat ialah perkara biasa, malah jurutera pun selalu jumpa ralat!



Ikut **gelung baiki** ini:

1 · Salin
Tekan **Salin untuk Gemini**. Kod dan mesej ralat disalin bersama.

2 · Tampal
Tampal dalam Gemini (Ctrl+V) dan hantar. Gemini akan menerangkan dan membetulkan.

3 · Cuba lagi
Salin kod baharu, tampal dalam `sketch.ino`, tekan F5 sekali lagi.

Kod berjalan tetapi **tidak buat apa yang kamu mahu**? Beritahu Gemini dengan jelas apa yang **berlaku** dan apa yang **sepatutnya berlaku**:

> Kod berjalan, tetapi LED menyala terus dan tidak padam. Sepatutnya LED menyala 3 saat sahaja selepas butang ditekan, kemudian padam. Tolong betulkan. Jangan ubah nombor pin.

Aktiviti: Fahami mesej ralat

Mesej ralat dalam Bahasa Inggeris. Ini tiga yang paling biasa:

Mesej	Maksud
`expected ';' before ...`	Terlupa titik koma `;` di hujung baris.
`'LED' was not declared in this scope`	Nama `LED` digunakan tetapi belum diberitahu apa ia.
`No such file or directory`	Kod perlukan pustaka yang tiada.

✎ AKTIVITI 9 · APA MAKSUDNYA?

10. sketch.ino:5:3: error: 'buzzer' was not declared in this scope

- □ A. Buzzer rosak
- □ B. Nama buzzer belum diberitahu dalam kod
- □ C. Pin buzzer salah
- □ D. Wi-Fi terputus

11. sketch.ino:9:21: error: expected ';' before '}' token

- □ A. Terlupa titik koma
- □ B. Terlupa kurungan
- □ C. Papan salah
- □ D. Perintang terlalu besar

SOALAN SETERUSNYA

Kita sudah boleh sambung, minta dan jalankan. Masa untuk membina rumah pintar Oyen!

BAB 7

Projek Rumah Pintar

Projek Rumah Pintar Oyen

Pilih projek mengikut tahap. Untuk setiap projek: **Pelan Pin → litar → prompt → jalankan → baiki**.

TAHAP 1

Lampu berkelip

LED rumah Oyen berkelip setiap 1 saat.

LED + perintang · Uno pin 13 atau ESP32 GPIO 26

TAHAP 1

Lampu isyarat jalan Oyen

Hijau 5 s → kuning 2 s → merah 5 s, berulang.

3 LED + 3 perintang

TAHAP 2

Loceng pintu Oyen

Tekan butang → buzzer berbunyi dan LED menyala 1 saat.

Butang, buzzer, LED

TAHAP 2

Lampu malam automatik

Apabila gelap, LED menyala. Apabila terang, LED padam.

Modul LDR + LED

TAHAP 3

Penggera panas

Jika suhu lebih 30 °C, buzzer berbunyi dan suhu dicetak ke Serial.

DHT22 + buzzer

TAHAP 3 · ESP32

“Oyen dah balik!”

Tekan butang → ESP32 hantar mesej melalui Wi-Fi ke papan kelas.

Butang + Wi-Fi (halaman seterusnya)

Tahap 3: “Oyen dah balik!” melalui Wi-Fi

Oyen masuk rumah dan memijak butang. Di skrin kelas, muncul mesej: **“Oyen dah balik! ”**

Dalam simulator, ESP32 bersambung ke Wi-Fi bernama **Wokwi-GUEST** (tiada kata laluan). Dari situ ia boleh sampai ke Internet sebenar.



Tambah bahan ini dalam prompt kamu:

> Sambung ke Wi-Fi "Wokwi-GUEST" tanpa kata laluan, guna WiFi.begin("Wokwi-GUEST", "", 6). Apabila butang ditekan, hantar HTTP POST ke [alamat pelayan kelas daripada cikgu] dengan teks "Oyen dah balik!" dan nama kumpulan saya, "Kumpulan Harimau". Cetak kod jawapan HTTP ke Serial Monitor.

Jangan hantar maklumat peribadi. Gunakan nama kumpulan, bukan nama penuh. Mesej melalui Wi-Fi simulator boleh dilihat oleh pengendali Wokwi.

Cikgu akan memberi alamat pelayan kelas. Tanpa alamat itu, kamu boleh menguji dengan `http://example.com/`: jika Serial menunjukkan **200**, ESP32 kamu sudah sampai ke Internet!

Senarai semak projek

✎ AKTIVITI 10 · TANDAKAN APABILA SIAP

- ☐ Saya menulis Pelan Pin.
- ☐ Litar dalam `diagram.json` ikut Pelan Pin.
- ☐ Prompt saya ada kelima-lima bahan.
- ☐ Saya semak nombor pin dalam kod Gemini.
- ☐ Kod berjaya dibina (tiada ralat).
- ☐ Simulator buat apa yang saya mahu.
- ☐ Saya boleh terangkan projek saya kepada kawan.

Penutup

Tahniah, jurutera cilik!

Rumah Oyen kini ada lampu, loceng, dan boleh memberitahu Mia apabila Oyen pulang. Mia tidak menulis kod sendiri, tetapi Mia tahu **apa yang mahu dibina** dan **bagaimana memintanya**. Itulah kemahiran yang paling penting!



Apa yang kita pelajari

- ✔ Arduino dan ESP32 ialah otak kecil yang ikut arahan kod.
- ✔ Pin ialah pintu: **output** untuk menghantar, **input** untuk menerima.
- ✔ LED perlu perintang dan arah yang betul. Butang ke GND dengan INPUT_PULLUP.
- ✔ **Pelan Pin** menghubungkan litar dengan prompt.
- ✔ Prompt hebat ada lima bahan: **Papan, Pelan Pin, Tugas, Peraturan, Semakan**.
- ✔ Ralat itu biasa: **Salin** → **Tampal** → **Cuba lagi**.

LAMPIRAN A

Nota cikgu

Nota cikgu

SEBELUM BENGKEL

- Pasang **MakmalArduinoMaya-Setup.exe** pada setiap PC (tidak perlu hak admin, kira-kira 5 minit, 2.2 GB).
- Aktifkan lesen Wokwi pada setiap PC (butang). Lesen ikut akaun Wokwi dan perlu diperbaharui lebih kurang sebulan sekali. Semak terma Wokwi untuk kegunaan kelas.
- Pastikan Internet berfungsi untuk Gemini dan simulator. Uji satu PC: buka Makmal ESP32, tekan F5, LED mesti berkelip.
- Sediakan fail `diagram.json` untuk projek Tahap 1 hingga 3 jika murid tidak akan melukis litar sendiri.
- Pastikan akaun Gemini murid (DELIMa) boleh digunakan. Gemini untuk murid bawah 18 tahun bergantung pada tetapan pentadbir sekolah.

SEMASA BENGKEL

- Bahagian 1 hingga 4 boleh diajar tanpa komputer: guna kad komponen dan Pelan Pin bercetak.
- Galakkan murid membaca kod Gemini sekali imbas: cari nombor pin, `delay()` dan `if`. Mereka tidak perlu faham semua.
- Jika kod Gemini guna `ledcAttach()` (versi 3.x) dan gagal dibina, minta Gemini "guna ESP32 Arduino core 2.0.17".
- Kerja berpasangan: seorang "jurutera litar", seorang "penulis prompt", bertukar setiap projek.

ANDAIAN MODUL INI

Ditulis untuk murid Tahun 4 hingga 6, bengkel sehari (kira-kira 4 jam 20 minit tanpa rehat). Pelan Pin dan prompt disimpan dalam pelayar murid sahaja. Tiada data dihantar ke mana-mana.

Contoh kod rujukan

Kod di bawah sudah diuji dengan pengkompil Makmal Arduino Maya. Guna untuk menyemak jika kod Gemini murid terlalu jauh menyimpang.

Loceng pintu Oyen (ESP32)

```
const int BUTANG = 4;    // butang ke GND
const int BUZZER = 25;
const int LED = 26;

void setup() {
  Serial.begin(115200);
  pinMode(BUTANG, INPUT_PULLUP);
  pinMode(LED, OUTPUT);
}

void loop() {
  if (digitalRead(BUTANG) == LOW) {   // ditekan
    Serial.println("Butang ditekan!");
    digitalWrite(LED, HIGH);
    tone(BUZZER, 1000);
    delay(1000);
    noTone(BUZZER);
    digitalWrite(LED, LOW);
  }
}
```

“Oyen dah balik!” (ESP32, Wi-Fi)

```
#include <WiFi.h>
#include <HTTPClient.h>

const int BUTANG = 4;
const char* PELAYAN = "http://example.com/";  // ganti dengan alamat pelayan kelas

void setup() {
  Serial.begin(115200);
  pinMode(BUTANG, INPUT_PULLUP);
  WiFi.begin("Wokwi-GUEST", "", 6);
  while (WiFi.status() != WL_CONNECTED) { delay(200); Serial.print("."); }
  Serial.println(" Wi-Fi bersambung!");
}

void loop() {
  if (digitalRead(BUTANG) == LOW) {
    HTTPClient http;
    http.begin(PELAYAN);
    http.addHeader("Content-Type", "text/plain");
    int kod = http.POST("Kumpulan Harimau: Oyen dah balik!");
    Serial.print("Jawapan pelayan: ");
    Serial.println(kod);
    http.end();
    delay(1000);   // elak hantar berkali-kali
  }
}
```

LAMPIRAN B

Jawapan aktiviti

1. B. Salah (Aktiviti 1 · Betul atau salah?)
Arduino hanya ikut arahan dalam kod. Tanpa kod, ia tidak buat apa-apa.

2. A. Betul (Aktiviti 1 · Betul atau salah?)
Ya! Itulah kelebihan ESP32 berbanding Arduino Uno.

3. C. LED terbalik (Aktiviti 3 · Uji sambungan)
Kaki panjang (+) mesti ke arah pin, kaki pendek (–) ke GND.

4. B. LOW (Aktiviti 3 · Uji sambungan)
Dengan INPUT_PULLUP, tekan = LOW. Ia terbalik!

5. D. GPIO 26 (Aktiviti 3 · Uji sambungan)
GPIO 6 hingga 11 untuk memori, GPIO 0 untuk butang BOOT, GPIO 34 input sahaja. GPIO 26 selamat.

6. C. Tugas (Aktiviti 5 · Detektif prompt)
Gemini tidak tahu apa LED itu perlu buat: berkelip? menyala terus? Tugas hilang.

7. B. Pelan Pin (Aktiviti 5 · Detektif prompt)
Butang dan buzzer di pin berapa? Gemini akan meneka pin, dan mungkin tidak sama dengan litar kita.

8. A. Papan (Aktiviti 5 · Detektif prompt)
Papan apa? Kod untuk ESP32 dan Uno ada sedikit perbezaan.

9. C. Apabila butang ditekan, buzzer berbunyi 1000 Hz selama 0.5 saat, kemudian senyap. Jika butang ditekan lagi, ulang. (Aktiviti 6 · Pilih tugas yang paling jelas)
Ia ada syarat (apabila ditekan), nilai (1000 Hz), masa (0.5 saat) dan apa berlaku selepas itu.

10. B. Nama buzzer belum diberitahu dalam kod (Aktiviti 9 · Apa maksudnya?)
‘not declared’ bermaksud nama itu belum ditakrifkan, contohnya terlupa baris const int buzzer = 25;

11. A. Terlupa titik koma (Aktiviti 9 · Apa maksudnya?)
‘expected ;’ bermaksud satu baris tiada titik koma di hujungnya.

LAMPIRAN C

Glosari

anod
: Kaki positif LED. Ia kaki yang lebih panjang dan disambung ke arah pin.

katod
: Kaki negatif LED. Ia kaki yang lebih pendek dan disambung ke GND.

kod
: Senarai arahan yang ditulis untuk komputer. Dalam Arduino, kod ditulis dalam bahasa C++.

mikropengawal
: Komputer yang sangat kecil di atas satu papan. Ia menjalankan satu program sahaja, berulang-ulang.

perintang
: Komponen yang mengurangkan arus elektrik. Nilainya dalam ohm (Ω). Untuk LED, kita biasa guna 220 Ω.

pin GPIO
: General Purpose Input/Output: pin serba guna. Kita boleh tetapkan sama ada pin itu untuk input (menerima) atau output (menghantar).

prompt
: Arahan atau soalan yang kita taip kepada AI seperti Gemini. Prompt yang jelas memberi jawapan yang lebih tepat.

pustaka
: Kod siap yang ditulis orang lain untuk memudahkan kita, contohnya untuk membaca sensor. Makmal Arduino Maya sudah ada pustaka DHT, LCD, servo dan lain-lain.

simulator
: Program komputer yang meniru benda sebenar. Simulator Wokwi meniru papan Arduino, lampu dan butang di skrin.