BENGKEL 2 HARI · SEKOLAH MENENGAH

Kebun Pintar Sekolah

Elektronik, ESP32 dan IoT, dibina bersama Gemini: dari pin pertama hingga papan pemuka di awan.

15 tahun ke atas · 2 hari × 4.5 jam · ESP32 · Gemini · IoT · MQTT



Kandungan

Pengenalan 4
Dua hari, satu kebun 5

Bab 1 · Sistem terbenam 6
Komputer yang tersembunyi di mana-mana 6
Bagaimana program Arduino berjalan 6

Bab 2 · Elektrik asas 8
Voltan, arus dan rintangan 8
Mengira perintang untuk LED 8

Bab 3 · Digital, analog, PWM 10
Isyarat digital dan perintang tarik-naik 10
Isyarat analog: membaca nilai di antara 10
PWM: “analog palsu” untuk memalapkan 11

Bab 4 · Sensor & aktuator 12
Komponen Kebun Pintar 12
Geganti dan I2C 13
Aktiviti: Pelan Pin kebun 14

Bab 5 · Kejuruteraan prompt 16
Prompt ialah spesifikasi 16
Menulis keperluan yang boleh diuji 16
Kekangan: menghalang tabiat buruk AI 17
Teknik prompt yang digunakan jurutera 18
Aktiviti: Pembina Spesifikasi 19

Bab 6 · Aliran kerja Makmal 20
Aliran kerja Makmal Arduino Maya 20
Membaca ralat compile 20

Bab 7 · Menyemak kod AI 22
Kod berjalan ≠ kod betul 22
Aktiviti: Cari pepijat 22
Corak millis() yang patut kamu kenal 23

Bab 8 · Nyahpepijat 25
Nyahpepijat seperti saintis 25

Bab 9 · IoT 27
Internet of Things 27
Dua cara bercakap: HTTP dan MQTT 27
Merancang topik MQTT 28

Bab 10 · AI secara bertanggungjawab 29
Kamu jurutera, AI pembantu 29

Bab 11 · Projek akhir 30
Projek akhir: Kebun Pintar Sekolah 30
Hasil kerja dan rubrik 31
Jadual ujian pasukan 32

Penutup 33
Apa yang kamu bawa pulang 33

Lampiran A · Nota cikgu 34

Lampiran B · Jawapan aktiviti 38

Lampiran C · Glosari 40

Pengenalan

> **Nurul** dan **Aiman** ahli Kelab STEM. Kebun sayur sekolah selalu layu semasa cuti hujung minggu kerana tiada siapa menyiram. Cikgu mencabar mereka: **bina kebun yang boleh menjaga dirinya sendiri**, dan laporkan keadaannya ke telefon.

Dalam dua hari, kamu akan belajar menjadi **jurutera yang mengarah AI**: kamu tentukan perkakasan, keperluan dan ujian. Gemini menulis kod. Kamu menyemak, menguji dan bertanggungjawab ke atas hasilnya.

Dua hari, satu kebun

Hari	Bahagian	Soalan yang dijawab	Masa
1	1 Sistem terbenam	Apakah ESP32 dan bagaimana ia berfikir?	25 min
1	2 Elektrik asas	Berapa ohm perintang untuk LED, dan kenapa?	40 min
1	3 Digital, analog, PWM	Bagaimana membaca sensor dan memalapkan lampu?	45 min
1	4 Sensor & aktuator	Apa yang kebun perlukan, dan di pin mana?	40 min
1	5 Kejuruteraan prompt	Bagaimana menulis spesifikasi yang Gemini tidak boleh salah faham?	60 min
1	6 Aliran kerja Makmal	Bagaimana membina, menjalankan dan membaca ralat?	40 min
2	7 Menyemak kod AI	Kod berjalan, tetapi betulkah ia?	45 min
2	8 Nyahpepijat	Bagaimana mencari punca masalah dengan sistematik?	30 min
2	9 IoT: Wi-Fi, HTTP, MQTT	Bagaimana kebun melapor ke telefon?	60 min
2	10 AI secara bertanggungjawab	Apa tanggungjawab kita apabila AI menulis kod?	15 min
2	11 Projek akhir	Bolehkah pasukan kamu membina Kebun Pintar?	2 jam

Masa ialah anggaran tanpa rehat. Cikgu boleh mengubah suai mengikut kelas.

BAB 1

Sistem terbenam

Komputer yang tersembunyi di mana-mana

Telefon kamu ada pemproses berbilang teras. Tetapi pengawal penyaman udara, mesin basuh dan lampu isyarat hanya perlukan komputer kecil yang murah dan jimat tenaga: **mikropengawal**.

INPUT

Sensor mengukur dunia: suhu, kelembapan tanah, cahaya, butang.

PROSES

Kod membuat keputusan: “jika tanah kering, siram”.

OUTPUT

Aktuator bertindak: pam air, LED, paparan LCD, mesej Wi-Fi.

	Arduino Uno	ESP32 DevKit
Pemproses	8-bit, 16 MHz	32-bit dwi-teras, 240 MHz
Voltan logik	5 V	**3.3 V**
Analog (ADC)	10-bit (0 hingga 1023)	12-bit (0 hingga 4095)
Wi-Fi / Bluetooth	Tiada	Ada

Untuk kebun yang melapor ke telefon, kita pilih **ESP32**.

Bagaimana program Arduino berjalan

Setiap program Arduino (dipanggil **sketch**) ada dua bahagian wajib:

```
void setup() {
  // berjalan SEKALI semasa papan dihidupkan
  // contoh: tetapkan pin, mulakan Serial, sambung Wi-Fi
}

void loop() {
  // berjalan BERULANG-ULANG selama-lamanya, beribu kali sesaat
  // contoh: baca sensor, buat keputusan, kawal pam
}
```

Kamu tidak perlu menulis kod ini sendiri hari ini, tetapi kamu **mesti boleh membacanya** untuk menyemak kerja Gemini.

✎ AKTIVITI 1 · SEMAK KEFAHAMAN

1. Di manakah sepatutnya Wi-Fi disambungkan pada permulaan?

- □ A. setup()
- □ B. loop()
- □ C. Di luar kedua-duanya
- □ D. Di dalam Serial Monitor

2. Sensor suhu ESP32 memberi isyarat 3.3 V. Jika disambung ke Uno (5 V), apa yang paling tepat?

- □ A. Tidak boleh langsung
- □ B. Pasti merosakkan Uno
- □ C. Biasanya boleh dibaca, tetapi isyarat 5 V ke ESP32 boleh merosakkan ESP32
- □ D. Tiada beza

SOALAN SETERUSNYA

ESP32 bekerja pada 3.3 V. Bagaimana kita memastikan arus yang mengalir ke LED dan sensor selamat?

BAB 2

Elektrik asas

Voltan, arus dan rintangan



Bayangkan paip air di kebun:

- **Voltan (V)** ialah tekanan air, iaitu daya yang menolak.
- **Arus (I)** ialah banyaknya air yang mengalir setiap saat.
- **Rintangan (R)** ialah paip yang sempit, yang menghadkan aliran.

Ketiga-tiganya dihubungkan oleh **Hukum Ohm**:

$$V = I \times R \qquad I = \frac{V}{R} \qquad R = \frac{V}{I}$$

V dalam volt, I dalam ampere, R dalam ohm

> **Had keselamatan pin ESP32:** pastikan arus setiap pin sekitar **10 hingga 20 mA**. Pin tidak boleh membekalkan arus besar seperti untuk motor atau pam. Untuk itu kita guna **geganti** (relay) atau pemacu motor.

Mengira perintang untuk LED

LED "memakan" sebahagian voltan, dipanggil **voltan hadapan** (V_f). Baki voltan jatuh pada perintang:

$$R = \frac{V_{bekalan} - V_f}{I}$$

LED merah $V_f \approx 2.0$ V · biru/putih ≈ 3.0 V

Contoh: ESP32 (3.3 V), LED merah (2.0 V), arus 10 mA = 0.010 A:

$$R = \frac{3.3 - 2.0}{0.010} = \frac{1.3}{0.010} = \mathbf{130\ \Omega}$$

Perintang dijual dalam nilai piawai. Kita pilih nilai piawai yang **sama atau lebih besar**: 150 Ω atau 220 Ω (arus lebih kecil, lebih selamat, LED masih cukup terang).

✎ AKTIVITI 2 · KALKULATOR PERINTANG

Kira perintang untuk setiap kes. Tunjukkan jalan kira sebagai pecahan.

Bekalan	LED	Arus	R dikira	Nilai piawai dipilih
5 V	Merah (2.0 V)	15 mA		
3.3 V	Hijau (2.2 V)	5 mA		
5 V	Biru (3.0 V)	10 mA		

3. Bekalan 5 V, LED merah (2.0 V), arus 15 mA. Berapakah R?

- □ A. 133 Ω
- □ B. 200 Ω
- □ C. 333 Ω
- □ D. 3000 Ω

4. Kalkulator memberi 130 Ω. Nilai piawai mana paling sesuai?

- □ A. 100 Ω
- □ B. 120 Ω
- □ C. 150 Ω
- □ D. 10 Ω

SOALAN SETERUSNYA

LED boleh dihidupkan dan dimatikan. Tetapi bagaimana dengan membaca kelembapan tanah, atau memalapkan lampu?

BAB 3

Digital, analog, PWM

Isyarat digital dan perintang tarik-naik

Isyarat **digital** hanya ada dua keadaan: **HIGH** (3.3 V) atau **LOW** (0 V). Butang ialah input digital.

Pin input yang tidak disambung ke mana-mana akan "terapung" dan membaca nilai rawak. Penyelesaiannya ialah **perintang tarik-naik**. ESP32 ada satu di dalamnya:

```
pinMode(BUTANG, INPUT_PULLUP);   // butang antara pin dan GND
if (digitalRead(BUTANG) == LOW) { /* ditekan */ }
```

Lantunan (bounce): apabila butang ditekan, logam di dalamnya melantun beberapa milisaat, jadi satu tekanan boleh dibaca sebagai 5 tekanan! Minta Gemini menambah **nyahlantun** (debounce) sekitar 50 ms.

Isyarat analog: membaca nilai di antara

Sensor kelembapan tanah dan sensor cahaya memberi voltan di antara 0 dan 3.3 V. **ADC** menukar voltan itu kepada nombor:

$$\text{bacaan} = \frac{V_{\text{masuk}}}{3.3\ \text{V}} \times 4095 \qquad V = \frac{\text{bacaan}}{4095} \times 3.3\ \text{V}$$

ESP32, ADC 12-bit

✎ AKTIVITI 3 · PENUKAR ADC

5. ESP32 membaca analogRead() = 1024. Lebih kurang berapakah voltannya?

- ☐ A. 0.33 V
- ☐ B. 0.83 V
- ☐ C. 1.65 V
- ☐ D. 3.3 V

Penting untuk IoT: apabila Wi-Fi digunakan, pin ADC2 (contohnya GPIO 25, 26, 27) **tidak boleh** membaca analog. Guna pin **ADC1: GPIO 32, 33, 34, 35, 36, 39** untuk sensor analog.

PWM: “analog palsu” untuk memalapkan

Pin digital hanya boleh HIGH atau LOW. Untuk memalapkan LED atau mengawal kelajuan kipas, kita tukar HIGH dan LOW dengan sangat laju. Peratus masa HIGH dipanggil **kitar tugas** (duty cycle).



$$\text{kitar tugas} = \frac{\text{masa HIGH}}{\text{tempoh}} \times 100\%$$

$$\text{nilai 8-bit} = \frac{\text{kitar tugas}}{100\%} \times 255$$

✎ AKTIVITI 4 · LIHAT PWM

6. LED dipalapkan pada nilai 191 (daripada 255). Lebih kurang berapakah kitar tugasnya?

- □ A. 25%
- □ B. 50%
- □ C. 75%
- □ D. 100%

Dalam ESP32 core 2.0.17: `ledcSetup(saluran, frekuensi, resolusi)` dan `ledcAttachPin(pin, saluran)`, kemudian `ledcWrite(saluran, nilai)`. Nyatakan versi core dalam prompt supaya Gemini guna fungsi yang betul.

SOALAN SETERUSNYA

Kita tahu jenis-jenis isyarat. Sekarang, komponen apa yang kebun perlukan, dan di pin mana?

BAB 4

Sensor & aktuator

Komponen Kebun Pintar



Komponen	Jenis	Sambungan	Dalam Wokwi
DHT22 suhu & kelembapan udara	INPUT digital	VCC 3V3, GND, SDA → GPIO 15	wokwi-dht22
Sensor kelembapan tanah	INPUT analog	AO → GPIO 34	Ganti dengan **potensiometer** (Wokwi tiada sensor tanah)
Modul LDR (cahaya)	INPUT analog	AO → GPIO 35	wokwi-photoresistor-sensor
Butang siram manual	INPUT digital	GPIO 4 ↔ GND, INPUT_PULLUP	wokwi-pushbutton
Modul geganti (pam)	OUTPUT	IN → GPIO 26	wokwi-relay-module
LCD 16×2 I2C	OUTPUT I2C	SDA → GPIO 21, SCL → GPIO 22, alamat 0x27	wokwi-lcd1602 (pins: i2c)
Buzzer amaran	OUTPUT	+ → GPIO 25	wokwi-buzzer

Geganti dan I2C

Geganti (relay)

Suis yang dikawal elektrik. Pin ESP32 yang lemah menghidupkan suis, dan suis itu menyambungkan bekalan pam yang lebih kuat.

Keselamatan: dalam bengkel, kita hanya guna pam kecil **5 V hingga 12 V**. **Jangan sekali-kali** menyambung bekalan elektrik rumah (240 V) tanpa pengawasan juruteknik bertauliah.

I2C: dua wayar, banyak peranti

I2C menggunakan dua wayar sahaja: **SDA** (data) dan **SCL** (jam). Banyak peranti boleh berkongsi dua wayar ini kerana setiap satu ada **alamat** (contohnya LCD = 0x27).

Pada ESP32: SDA = GPIO 21, SCL = GPIO 22. Jangan guna pin ini untuk benda lain.



Aktiviti: Pelan Pin kebun

Jurutera tidak menyambung wayar secara rawak. Mereka menulis **Pelan Pin** dahulu, iaitu satu jadual yang menjadi kontrak antara litar, prompt dan kod.

✎ **AKTIVITI 5 · PELAN PIN ESP32**

Komponen	Pin	Jenis	Nota (voltan, alamat, ADC1?)
DHT22			
Sensor tanah (potensiometer)			
LDR			
Butang siram			
Geganti pam			
LCD I2C (SDA/SCL)			
Buzzer			

Litar siap untuk Wokwi: kebun-diagram.json (salin ke `diagram.json` projek).

SOALAN SETERUSNYA

Pelan Pin sudah ada. Bagaimana kita menukarnya kepada arahan yang Gemini tidak boleh salah faham?

BAB 5

Kejuruteraan prompt

Prompt ialah spesifikasi



Gemini menulis kod berdasarkan **apa yang kamu tulis sahaja**. Ia tidak nampak litar kamu, tidak tahu versi pustaka kamu, dan akan **meneka** apa yang tidak dinyatakan. Setiap tekaan ialah peluang untuk pepijat.

Jurutera menulis **spesifikasi**. Prompt yang baik ialah spesifikasi yang ringkas, dengan lima bahagian:

1 · KONTEKS
Peranan dan projek. Siapa pengguna?

2 · PERKAKASAN
Papan, versi core, Pelan Pin penuh.

3 · KEPERLUAN
Bernombor, boleh diuji, dengan nilai dan masa.

4 · KEKANGAN
Pustaka, gaya kod, tiada delay() panjang.

5 · UJIAN
Output Serial yang dijangka untuk setiap keadaan.

Menulis keperluan yang boleh diuji

Keperluan yang baik boleh dijawab dengan **lulus** atau **gagal**. Elakkan perkataan kabur seperti “cepat”, “kering”, “kadang-kadang”.

Kabur	Boleh diuji
Siram bila tanah kering.	K1. Jika bacaan tanah (GPIO 34) > 3000, hidupkan geganti (GPIO 26) selama 5 saat, kemudian tunggu sekurang-kurangnya 60 saat sebelum menyiram semula.
Tunjuk suhu.	K2. Setiap 2 saat, papar “S:28.5C L:65%” pada baris 1 LCD (1 titik perpuluhan).
Beri amaran bila panas.	K3. Jika suhu ≥ 35.0 °C, buzzer berbunyi 2000 Hz 200 ms setiap 5 saat sehingga suhu < 34.0 °C.

Perhatikan K3 menggunakan dua nilai (35.0 dan 34.0). Ini dipanggil **histeresis**, iaitu ia mengelakkan buzzer berbunyi-berhenti tanpa henti apabila suhu berada di sekitar 35 °C.

✎ AKTIVITI 6 · KABUR ATAU BOLEH DIUJI?

7. “Lampu LED menyala apabila gelap.”

- □ A. Kabur: apa maksud gelap?
- □ B. Boleh diuji

8. “Setiap 10 saat, terbitkan JSON {"suhu":x,"tanah":y} ke topik MQTT makmal/smk01/k3/kebun.”

- □ A. Kabur
- □ B. Boleh diuji

Kekangan: menghalang tabiat buruk AI

Tanpa kekangan, kod AI selalunya “berjalan” tetapi rapuh. Kekangan yang berguna untuk projek kita:

- ✔ **Versi:** “ESP32 Arduino core 2.0.17. Guna ledcSetup/ledcAttachPin untuk PWM.”
- ✔ **Pustaka:** “Guna hanya pustaka ini: DHT sensor library, LiquidCrystal_I2C, PubSubClient, ArduinoJson v7.”
- ✔ **Tidak menyekat:** “Jangan guna delay() lebih 50 ms dalam loop(). Guna millis() untuk pemasaan.”
- ✔ **Kebolehbacaan:** “Semua pin dan ambang sebagai pemalar bernama di bahagian atas.”
- ✔ **Ketahanan:** “Semak jika bacaan DHT ialah NaN dan jangan guna bacaan itu.”
- ✔ **Format jawapan:** “Beri kod penuh dalam satu blok. Selepas kod, terangkan dalam 5 baris bagaimana ia memenuhi K1 hingga K3.”

Kenapa millis()? `delay(5000)` membekukan papan selama 5 saat: butang tidak dibaca, LCD tidak dikemas kini, MQTT terputus. Dengan **millis()**, program menyemak jam dan terus bekerja.

Teknik prompt yang digunakan jurutera

Teknik	Contoh ayat	Bila digunakan
Rancang dahulu	"Sebelum menulis kod, senaraikan langkah dan fungsi yang akan kamu buat. Tunggu saya setuju."	Projek besar
Minta soalan	"Jika ada apa-apa yang tidak jelas dalam spesifikasi, tanya saya dahulu."	Sebelum kod pertama
Satu perubahan	"Kod sedia ada berfungsi. Tambah K4 sahaja. Jangan ubah bahagian lain."	Menambah ciri
Ralat + kod	"Ini ralat compile dan kod penuh. Terangkan punca, kemudian beri kod yang dibetulkan."	Ralat compile
Dijangka vs berlaku	"Dijangka: pam hidup 5 s. Berlaku: pam hidup terus. Output Serial: …"	Pepijat logik
Terangkan	"Terangkan baris 40 hingga 55 seolah-olah kepada pelajar Tingkatan 4."	Memahami kod

✎ **AKTIVITI 7 · PILIH TEKNIK**

9. Kod kebun sudah berfungsi. Kamu mahu menambah paparan cahaya pada LCD. Prompt mana paling baik?

□ A. "Tulis semula kod kebun dengan LDR."

□ B. "Tambah LDR."

□ C. "Kod di bawah berfungsi. Tambah K5: papar bacaan LDR (GPIO 35) di baris 2 LCD setiap 2 s. Jangan ubah bahagian lain."

□ D. "Kenapa LCD tak papar cahaya?"

Aktiviti: Pembina Spesifikasi

✎ AKTIVITI 8 · BINA PROMPT SPESIFIKASI

KONTEKS: Saya pelajar sekolah menengah membina

_____

PERKAKASAN: ESP32 DevKit (ESP32 Arduino core 2.0.17). Sambungan:
- _____ di GPIO _____ · - _____ di GPIO _____
- _____ di GPIO _____ · - _____ di GPIO _____

KEPERLUAN:
K1.

_____
K2.

_____
K3.

_____

KEKANGAN: core 2.0.17 · pustaka yang dibenarkan sahaja · tiada delay() panjang, guna millis() · pemalar bernama · semak NaN · kod penuh satu blok, komen BM, jangan ubah pin.

UJIAN PENERIMAAN:
U1.

_____
U2.

_____

SOALAN SETERUSNYA

Spesifikasi siap dan Gemini memberi kod. Bagaimana kita membina, menjalankan dan menguji kod itu?

BAB 6

Aliran kerja Makmal

Aliran kerja Makmal Arduino Maya

1. Buka **Makmal ESP32** dari Desktop. Untuk projek baharu, tekan **Projek Baharu**.
2. **Litar:** lukis di wokwi.com (jangan tekan Play di sana), salin tab `diagram.json` ke `diagram.json` projek. Atau guna fail litar yang cikgu beri.
3. **Kod:** tampal kod Gemini ke dalam `sketch.ino` (Ctrl+A, Ctrl+V).
4. ▶ **Bina & Jalankan** (F5). Simulator terbuka. Output Serial muncul di Terminal.
5. **Ralat?** Tekan **Salin untuk Gemini**, kemudian tampal dalam Gemini. Ralat dan kod disalin bersama.



Lesen Wokwi: kali pertama, tekan Lesen Wokwi → log masuk wokwi.com → GET YOUR LICENSE → benarkan pelayar membuka VS Code. Lesen perlu diperbaharui lebih kurang sebulan sekali. Simulasi perlukan Internet; pengkompilan tidak.

Dalam simulator, klik pada DHT22 atau potensiometer untuk menukar suhu dan kelembapan tanah semasa program berjalan. Ini cara terbaik untuk menguji K1 hingga K3.

Membaca ralat compile

Mesej ralat bermula dengan `sketch.ino:baris:lajur: error:`. Baca baris **pertama** dahulu, kerana ralat lain selalunya akibat yang pertama.

Mesej	Punca biasa	Tindakan
`'X' was not declared in this scope`	Nama belum ditakrif, atau salah ejaan	Semak ejaan; minta Gemini menambah definisi
`No such file or directory`	Pustaka tiada dalam Makmal	Minta Gemini guna pustaka yang disenaraikan
`'ledcAttach' was not declared`	Kod untuk core 3.x	“Guna core 2.0.17 (ledcSetup/ledcAttachPin)”
`expected ';' before`	Titik koma hilang pada baris sebelumnya	Lihat baris di **atas** nombor yang diberi
`redefinition of 'void setup()'`	Kod ditampal dua kali	Ctrl+A, padam, tampal sekali sahaja

✎ AKTIVITI 9 · DIAGNOSIS RALAT

10. fatal error: Adafruit_SSD1306_Wire.h: No such file or directory

- ☐ A. Wayar LCD terputus
- ☐ B. Kod meminta pustaka yang tiada; minta versi dengan pustaka yang disenaraikan
- ☐ C. Pin I2C salah
- ☐ D. Perlu Wi-Fi

11. error: 'ledcAttach' was not declared in this scope

- ☐ A. LED rosak
- ☐ B. Lupa titik koma
- ☐ C. Pin PWM salah
- ☐ D. Kod ditulis untuk ESP32 core 3.x; minta versi core 2.0.17

SOALAN SETERUSNYA

Kod berjaya dibina dan berjalan. Tetapi “berjalan” tidak bermaksud “betul”. Bagaimana kita menyemak kod AI?

BAB 7

Menyemak kod AI

Kod berjalan ≠ kod betul



AI boleh menghasilkan kod yang kelihatan meyakinkan tetapi mengandungi kesilapan halus. Sebelum menjalankan kod pada perkakasan sebenar (contohnya pam), lakukan **semakan kod**:

1. **Pin:** setiap pemalar pin sama dengan Pelan Pin?
2. **Keperluan:** setiap K1, K2, K3 ada dalam kod? Cari nilai ambang dan masa.
3. **Pemasaan:** ada `delay()` panjang yang membekukan program?
4. **Keadaan selamat:** apa berlaku jika sensor gagal (NaN)? Adakah pam boleh hidup tanpa henti?
5. **Nombor ajaib:** nilai seperti 3000 atau 5000 ditulis sebagai pemalar bernama?
6. **Faham:** bolehkah kamu terangkan setiap fungsi kepada rakan?

Aktiviti: Cari pepijat

Gemini memberi kod ini untuk K1 (“jika tanah > 3000, pam 5 saat, kemudian rehat 60 saat”). Pelan Pin: tanah GPIO 34, geganti GPIO 26.

```
const int TANAH = 34;
const int PAM = 27;
const int AMBANG = 3000;

void loop() {
  int t = analogRead(TANAH);
  if (t > AMBANG) {
    digitalWrite(PAM, HIGH);
    delay(5000);
    digitalWrite(PAM, LOW);
  }
}
```

✎ **AKTIVITI 10 · SEMAKAN KOD**

12. Pepijat 1: apa yang salah dengan pin?

- □ A. TANAH sepatutnya 26
- □ B. PAM sepatutnya 26, bukan 27
- □ C. AMBANG sepatutnya pin
- □ D. Tiada masalah pin

13. Pepijat 2: keperluan mana yang tiada?

- □ A. Bacaan tanah
- □ B. Pam 5 saat
- □ C. Rehat 60 saat sebelum menyiram semula
- □ D. Ambang 3000

14. Pepijat 3: kenapa delay(5000) bermasalah dalam projek penuh?

- □ A. Program beku 5 s: butang, LCD dan MQTT tidak dikemas kini
- □ B. delay() tidak wujud pada ESP32
- □ C. 5000 terlalu kecil
- □ D. Tiada masalah

Corak millis() yang patut kamu kenal

Kamu tidak perlu menulisnya, tetapi kenali bentuknya semasa menyemak kod Gemini:

```
unsigned long mulaPam = 0;
bool pamHidup = false;

void loop() {
  unsigned long kini = millis();
  if (!pamHidup && tanahKering() && kini - akhirSiram >= REHAT_MS) {
    pamHidup = true;  mulaPam = kini;  digitalWrite(PAM, HIGH);
  }
  if (pamHidup && kini - mulaPam >= SIRAM_MS) {
    pamHidup = false; akhirSiram = kini; digitalWrite(PAM, LOW);
  }
  // ...bahagian lain terus berjalan: LCD, butang, MQTT
}
```

Tanda-tanda kod yang baik: `kini - mula >= TEMPOH` (selamat walaupun millis() melimpah selepas 49 hari), dan pemboleh ubah **keadaan** seperti `pamHidup`. Ini dipanggil **mesin keadaan**.

✎ AKTIVITI 11 · PEMERIKSA PIN

Semakan	Lulus?	Catatan
Pin sama dengan Pelan Pin	☐	
Semua keperluan K1 hingga K3 ada	☐	
Tiada delay() panjang	☐	
Keadaan selamat (NaN, pam tidak hidup terus)	☐	
Pemalar bernama	☐	
Saya boleh terangkan kod	☐	

SOALAN SETERUSNYA

Semakan membantu, tetapi pepijat tetap berlaku. Bagaimana mencari puncanya dengan sistematik?

BAB 8

Nyahpepijat

Nyahpepijat seperti saintis

Nyahpepijat ialah kaedah saintifik: buat hipotesis, uji, perhatikan.

1 · Lihat dengan Serial

Minta Gemini menambah `Serial.printf()` pada setiap keputusan penting: nilai sensor, keadaan pam, masa.

2 · Bahagi dan takluk

Uji satu komponen pada satu masa. LCD tidak papar? Cuba program LCD sahaja dahulu.

3 · Ubah satu benda

Hanya satu perubahan antara setiap ujian, supaya kamu tahu perubahan mana yang membetulkan.

4 · Litar dahulu

Semak `diagram.json`: wayar ke pin betul? GND disambung? LCD guna alamat 0x27?

✎ AKTIVITI 12 · HIPOTESIS

15. LCD menyala tetapi kosong. Serial menunjukkan suhu dibaca dengan betul. Hipotesis paling munasabah?

- ☐ A. DHT22 rosak
- ☐ B. Alamat I2C atau pin SDA/SCL tidak sepadan
- ☐ C. Wi-Fi terputus
- ☐ D. Geganti salah

16. Serial menunjukkan “Suhu: nan”. Apa langkah pertama?

- ☐ A. Semak sambungan DHT22 (SDA ke GPIO 15, VCC, GND) dan pin dalam kod
- ☐ B. Tukar ESP32
- ☐ C. Buang semakan NaN
- ☐ D. Naikkan delay

SOALAN SETERUSNYA

Kebun sudah berfungsi di meja. Bagaimana ia melapor ke telefon Cikgu semasa cuti hujung minggu?

BAB 9

IoT

Internet of Things

IoT bermaksud benda fizikal yang bercakap melalui Internet. Kebun kita akan menghantar data ke **awan**, dan sesiapa yang dibenarkan boleh melihatnya.



Dalam simulator, ESP32 bersambung ke rangkaian **Wokwi-GUEST** (tiada kata laluan):

```
WiFi.begin("Wokwi-GUEST", "", 6);   // saluran 6 = sambungan lebih cepat
while (WiFi.status() != WL_CONNECTED) { delay(200); }
```

Privasi: trafik Wi-Fi simulator melalui get laluan awam Wokwi dan boleh dipantau. Jangan hantar nama, nombor telefon atau kata laluan sebenar.

Dua cara bercakap: HTTP dan MQTT

HTTP

Seperti menghantar surat dan menunggu balasan. ESP32 membuat **permintaan** (GET atau POST) ke pelayan web, dan pelayan membalas dengan kod (200 = OK).

Sesuai untuk laporan sekali-sekala.

MQTT

Seperti kumpulan WhatsApp. Peranti **menerbitkan** (publish) mesej ke **topik** di **broker**, dan sesiapa yang **melanggan** (subscribe) topik itu menerimanya serta-merta.

Ringan, sesuai untuk IoT dan data berterusan.

Data dihantar dalam format **JSON**:

```
{"suhu": 28.5, "lembap": 65, "tanah": 3120, "pam": false}
```

Merancang topik MQTT

Kita guna broker awam percuma `broker.hivemq.com` (port 1883). Kerana ia awam, **sesiapa sahaja** boleh membaca topik yang diteka. Oleh itu, guna topik yang unik dan jangan hantar data peribadi.

Struktur topik bengkel: `makmal/<kod sekolah>/<kumpulan>/<data>`

✎ **AKTIVITI 13 · PEREKA TOPIK**

17. Topik mana paling sesuai untuk kumpulan 3 menghantar data kebun?

- □ A. kebun
- □ B. test
- □ C. makmal/smk01/k3/kebun
- □ D. nurul.aiman.0123456789

Untuk melihat data: buka klien web MQTT (contohnya HiveMQ WebSocket Client), sambung ke broker.hivemq.com, dan langgan `makmal/smk01/k3/#`.

SOALAN SETERUSNYA

Kebun kini melapor ke awan, dengan kod yang ditulis oleh AI. Siapa yang bertanggungjawab jika ia silap?

BAB 10

AI secara bertanggungjawab

Kamu jurutera, AI pembantu



- ✔ **Kamu bertanggungjawab** ke atas kod yang kamu gunakan, walaupun AI yang menulisnya.
- ✔ **Faham sebelum guna:** jangan hantar kod yang kamu tidak boleh terangkan.
- ✔ **Jujur:** nyatakan “dibina dengan bantuan Gemini” dalam laporan dan pembentangan.
- ✔ **Privasi:** jangan masukkan data peribadi, kata laluan atau maklumat sekolah yang sulit ke dalam prompt.
- ✔ **Sahkan:** AI boleh salah dengan yakin. Uji setiap keperluan.
- ✔ **Keselamatan fizikal:** kod yang mengawal pam, motor atau pemanas mesti ada had masa dan keadaan selamat.

✎ **AKTIVITI 14 · SITUASI**

18. Rakan kamu mahu menampal kata laluan Wi-Fi rumah dan alamatnya ke dalam Gemini “supaya kod lengkap”. Nasihat terbaik?

□ A. Tidak mengapa, Gemini selamat

□ B. Guna nilai contoh seperti "NAMA_WIFI" dalam prompt, isi nilai sebenar sendiri dalam kod

□ C. Hantar melalui e-mel

□ D. Tulis dalam komen kod

BAB 11

Projek akhir

Projek akhir: Kebun Pintar Sekolah



Taklimat: bina sistem yang menjaga kebun sekolah semasa hujung minggu dan melapor keadaannya. Pasukan 3 hingga 4 orang, 2 jam.

Tahap	Keperluan
Minimum	DHT22 dan sensor tanah dibaca; LCD memaparkan suhu, kelembapan dan tanah; pam (geganti) disiram mengikut K1 dengan rehat.
Sasaran	+ Butang siram manual dengan nyahlantun; amaran suhu dengan histeresis; tiada delay() panjang.
Cabaran	+ Terbit JSON ke MQTT setiap 10 s; terima arahan “siram” melalui topik MQTT; LDR untuk tidak menyiram pada tengah hari terik.

Peranan pasukan: Jurutera Perkakasan (litar, Pelan Pin) · Jurutera Spesifikasi (prompt, keperluan) · Penguji (ujian, semakan kod) · Pembentang (demo, log).

Hasil kerja dan rubrik

1. **Pelan Pin** dan `diagram.json`.
2. **Log prompt:** setiap prompt yang digunakan, dan apa yang berubah selepasnya.
3. **Kod akhir** dengan komen.
4. **Jadual ujian:** setiap keperluan, langkah ujian, lulus/gagal.
5. **Demo 3 minit** dalam simulator, termasuk satu keadaan gagal yang dikendalikan dengan selamat.

Kriteria	Cemerlang (4)	Baik (3)	Asas (2)
Fungsi	Semua keperluan sasaran + satu cabaran	Semua keperluan sasaran	Keperluan minimum
Spesifikasi & prompt	Keperluan bernombor, kekangan dan ujian jelas; log lengkap	Kebanyakan keperluan boleh diuji	Prompt umum
Kualiti & keselamatan kod	millis(), pemalar bernama, NaN, had pam	Sebahagian ciri	Berjalan tetapi rapuh
Ujian	Setiap keperluan diuji dan direkod	Kebanyakan diuji	Ujian tidak direkod
Pembentangan	Jelas, menerangkan kod dan peranan AI	Jelas	Kurang jelas

Jadual ujian pasukan

✎ **AKTIVITI 15 · REKOD UJIAN**

Kep.	Langkah ujian dalam simulator	Dijangka	Lulus?
K1	Pusing potensiometer tanah melebihi 3000	“Pam ON”, geganti hidup 5 s, kemudian rehat 60 s	☐
K2			☐
K3			☐
K4			☐
K5			☐

Penutup

Apa yang kamu bawa pulang

- ✔ Hukum Ohm untuk mengira perintang, dan had arus pin ESP32.
- ✔ Digital, analog (ADC 12-bit, guna ADC1 dengan Wi-Fi) dan PWM.
- ✔ Pelan Pin sebagai kontrak antara litar, prompt dan kod.
- ✔ Prompt sebagai spesifikasi: konteks, perkakasan, keperluan, kekangan, ujian.
- ✔ Semak kod AI: pin, keperluan, millis(), keadaan selamat, kefahaman.
- ✔ IoT dengan Wi-Fi, HTTP, JSON dan MQTT, dengan privasi dalam fikiran.
- ✔ Kamu jurutera, AI pembantu. Tanggungjawab tetap pada kamu.

Seterusnya: cuba bina kebun ini dengan perkakasan sebenar. Kod yang sama boleh dimuat naik ke ESP32 sebenar.

LAMPIRAN A

Nota cikgu

Jadual dan persediaan

CADANGAN JADUAL

Hari 1		Hari 2	
08:30	Bahagian 1–2	08:30	Ulang kaji + Bahagian 7
09:35	Bahagian 3	09:20	Bahagian 8
10:20	Rehat	09:50	Bahagian 9
10:40	Bahagian 4	10:50	Rehat
11:20	Bahagian 5	11:10	Bahagian 10 + taklimat projek
12:20	Bahagian 6 + Makmal 1 (PWM + butang)	11:30	Projek akhir
13:00	Makmal 2 (DHT22 → LCD), tamat	13:30	Demo dan penutup

PERSEDIAAN

- Pasang Makmal Arduino Maya dan aktifkan lesen Wokwi pada setiap PC. Uji satu projek dengan Wi-Fi.
- Salin `litar/kebun-diagram.json` ke setiap PC atau kongsi melalui Google Classroom. Uji ia dalam Makmal sebelum kelas.
- Semak broker MQTT boleh dicapai dari makmal (port 1883 keluar). Jika disekat, guna HTTP sahaja.
- Pastikan akaun Gemini pelajar (DELIMa) aktif.

Kod rujukan: Kebun Pintar (tahap sasaran + MQTT)

Sudah diuji dengan pengkompil Makmal Arduino Maya (ESP32 core 2.0.17). Gunakan untuk menilai kod pasukan, bukan untuk diedarkan sebelum projek.

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>
#include <DHT.h>
#include <LiquidCrystal_I2C.h>

const int PIN_DHT = 15, PIN_TANAH = 34, PIN_LDR = 35;
const int PIN_BUTANG = 4, PIN_PAM = 26, PIN_BUZZER = 25;
const int AMBANG_TANAH = 3000;             // lebih besar = lebih kering
const float SUHU_PANAS = 35.0, SUHU_OK = 34.0;
const unsigned long SIRAM_MS = 5000, REHAT_MS = 60000;
const unsigned long BACA_MS = 2000, TERBIT_MS = 10000;
const char* TOPIK = "makmal/smk01/k3/kebun";

DHT dht(PIN_DHT, DHT22);
LiquidCrystal_I2C lcd(0x27, 16, 2);
WiFiClient wifi;
PubSubClient mqtt(wifi);

float suhu = NAN, lembap = NAN;
int tanah = 0;
bool pamHidup = false, amaran = false;
unsigned long mulaPam = 0, akhirSiram = 0, akhirBaca = 0, akhirTerbit = 0,
akhirBunyi = 0;
int butangLepas = HIGH; unsigned long masaButang = 0;

void mulaSiram(unsigned long kini) {
  pamHidup = true; mulaPam = kini; digitalWrite(PIN_PAM, HIGH);
  Serial.println("Pam ON");
}

void sambungMqtt() {
  if (mqtt.connected()) return;
  String id = "makmal-" + String((uint32_t)ESP.getEfuseMac(), HEX);
  if (mqtt.connect(id.c_str())) Serial.println("MQTT bersambung");
}

void setup() {
  Serial.begin(115200);
  pinMode(PIN_BUTANG, INPUT_PULLUP);
  pinMode(PIN_PAM, OUTPUT);
  dht.begin();
  lcd.init(); lcd.backlight();
  WiFi.begin("Wokwi-GUEST", "", 6);
  while (WiFi.status() != WL_CONNECTED) { delay(200); Serial.print("."); }
  mqtt.setServer("broker.hivemq.com", 1883);
  Serial.println("\nKebun sedia");
}

void loop() {
  unsigned long kini = millis();
  sambungMqtt();
```

```
  mqtt.loop();

  if (kini - akhirBaca >= BACA_MS) {                // K2: baca dan papar
    akhirBaca = kini;
    float s = dht.readTemperature(), l = dht.readHumidity();
    if (!isnan(s) && !isnan(l)) { suhu = s; lembap = l; }
    tanah = analogRead(PIN_TANAH);
    lcd.setCursor(0, 0); lcd.printf("S:%4.1fC L:%3.0f%% ", suhu, lembap);
    lcd.setCursor(0, 1); lcd.printf("Tanah:%4d %s ", tanah, pamHidup ? "PAM" : "
");
  }

  if (!pamHidup && tanah > AMBANG_TANAH && kini - akhirSiram >= REHAT_MS)
mulaSiram(kini);   // K1
  int b = digitalRead(PIN_BUTANG);                  // siram manual, nyahlantun 50
ms
  if (b != butangLepas && kini - masaButang > 50) {
    masaButang = kini; butangLepas = b;
    if (b == LOW && !pamHidup) mulaSiram(kini);
  }
  if (pamHidup && kini - mulaPam >= SIRAM_MS) {
    pamHidup = false; akhirSiram = kini; digitalWrite(PIN_PAM, LOW);
    Serial.println("Pam OFF");
  }

  if (!isnan(suhu)) {                               // K3: amaran dengan
histeresis
    if (suhu >= SUHU_PANAS) amaran = true;
    if (suhu < SUHU_OK) amaran = false;
  }
  if (amaran && kini - akhirBunyi >= 5000) { akhirBunyi = kini; tone(PIN_BUZZER,
2000, 200); }

  if (kini - akhirTerbit >= TERBIT_MS && mqtt.connected()) {   // cabaran: MQTT
    akhirTerbit = kini;
    JsonDocument doc;
    doc["suhu"] = suhu; doc["lembap"] = lembap; doc["tanah"] = tanah;
    doc["cahaya"] = analogRead(PIN_LDR); doc["pam"] = pamHidup;
    char buf[160]; serializeJson(doc, buf);
    mqtt.publish(TOPIK, buf);
    Serial.println(buf);
  }
}
```

LAMPIRAN B

Jawapan aktiviti

1. A. setup() (Aktiviti 1 · Semak kefahaman)
Sambungan Wi-Fi dibuat sekali sahaja semasa papan dihidupkan, jadi ia diletakkan dalam setup().

2. C. Biasanya boleh dibaca, tetapi isyarat 5 V ke ESP32 boleh merosakkan ESP32 (Aktiviti 1 · Semak kefahaman)
ESP32 ialah peranti 3.3 V. Isyarat 5 V ke pin ESP32 boleh merosakkannya. Sentiasa semak voltan logik.

3. B. 200 Ω (Aktiviti 2 · Kalkulator perintang)
R = (5 – 2.0) / 0.015 = 3 / 0.015 = 200 Ω.

4. C. 150 Ω (Aktiviti 2 · Kalkulator perintang)
Pilih nilai piawai yang sama atau lebih besar supaya arus tidak melebihi rancangan.

5. B. 0.83 V (Aktiviti 3 · Penukar ADC)
V = 1024/4095 × 3.3 ≈ 0.83 V.

6. C. 75% (Aktiviti 4 · Lihat PWM)
191/255 ≈ 0.75 = 75%.

7. A. Kabur: apa maksud gelap? (Aktiviti 6 · Kabur atau boleh diuji?)
Nyatakan ambang, contohnya: jika bacaan LDR (GPIO 35) > 2500.

8. B. Boleh diuji (Aktiviti 6 · Kabur atau boleh diuji?)
Ada selang masa, format data dan destinasi yang tepat.

9. C. “Kod di bawah berfungsi. Tambah K5: papar bacaan LDR (GPIO 35) di baris 2 LCD setiap 2 s. Jangan ubah bahagian lain.” (Aktiviti 7 · Pilih teknik)
Satu perubahan sahaja, dengan keperluan yang boleh diuji, dan arahan untuk tidak menyentuh kod yang berfungsi.

10. B. Kod meminta pustaka yang tiada; minta versi dengan pustaka yang disenaraikan (Aktiviti 9 · Diagnosis ralat)
‘No such file’ pada #include bermaksud pustaka itu tiada. Nyatakan senarai pustaka dalam kekangan.

11. D. Kod ditulis untuk ESP32 core 3.x; minta versi core 2.0.17 (Aktiviti 9 · Diagnosis ralat)
ledcAttach() hanya wujud dalam core 3.x. Makmal menggunakan 2.0.17.

12. B. PAM sepatutnya 26, bukan 27 (Aktiviti 10 · Semakan kod)
Pelan Pin menetapkan geganti di GPIO 26. Kod menggunakan 27, jadi pam tidak akan hidup.

13. C. Rehat 60 saat sebelum menyiram semula (Aktiviti 10 · Semakan kod)
Tanpa rehat 60 s, jika tanah masih kering, pam hidup semula serta-merta dan berulang-ulang, boleh membanjiri kebun.

14. A. Program beku 5 s: butang, LCD dan MQTT tidak dikemas kini (Aktiviti 10 · Semakan kod)
Guna millis() supaya pam dipantau tanpa membekukan bahagian lain.

15. B. Alamat I2C atau pin SDA/SCL tidak sepadan (Aktiviti 12 · Hipotesis)
Data sensor ada (Serial), jadi masalah pada laluan paparan: alamat 0x27, SDA 21, SCL 22.

16. A. Semak sambungan DHT22 (SDA ke GPIO 15, VCC, GND) dan pin dalam kod (Aktiviti 12 · Hipotesis)
nan bermaksud bacaan gagal. Punca biasa ialah wayar data atau pin yang salah.

17. C. makmal/smk01/k3/kebun (Aktiviti 13 · Pereka topik)
Unik, tersusun, dan tiada maklumat peribadi. ‘test’ dan ‘kebun’ terlalu umum, dan topik terakhir mendedahkan nombor telefon.

18. B. Guna nilai contoh seperti "NAMA_WIFI" dalam prompt, isi nilai sebenar sendiri dalam kod (Aktiviti 14 · Situasi)
Jangan kongsi rahsia dengan AI. Guna nilai contoh, kemudian isi sendiri secara setempat.

LAMPIRAN C

Glosari

ADC
: Analog-to-Digital Converter: penukar yang mengubah voltan kepada nombor. ESP32 ada ADC 12-bit, iaitu 4096 tahap.

Arus
: Kadar aliran cas elektrik. Unit: ampere (A). 1 mA = 0.001 A.

broker
: Pelayan MQTT yang menerima mesej daripada penerbit dan menghantarnya kepada semua pelanggan topik itu.

histeresis
: Jurang antara nilai hidup dan nilai mati supaya sistem tidak berkelip-kelip hidup/mati apabila bacaan berada betul-betul pada had.

I2C
: Inter-Integrated Circuit: bas komunikasi dua wayar (SDA data, SCL jam). Setiap peranti ada alamat sendiri, contohnya LCD 0x27.

IoT
: Internet of Things: peranti fizikal (sensor, alat) yang bersambung ke Internet untuk menghantar data atau menerima arahan.

JSON
: JavaScript Object Notation: format teks untuk data berstruktur, contohnya {"suhu":28.5}. Mudah dibaca manusia dan mesin.

kitar tugas
: Peratus masa isyarat PWM berada pada HIGH dalam satu kitaran. 0% = sentiasa mati, 100% = sentiasa hidup.

mesin keadaan
: Cara menyusun program sebagai beberapa keadaan (contohnya MENUNGGU, MENYIRAM, REHAT) dengan syarat yang jelas untuk bertukar keadaan.

mikropengawal
: Cip yang mengandungi pemproses, memori dan pin input/output dalam satu pakej. Ia menjalankan satu program secara berterusan.

millis()
: Fungsi yang memberi bilangan milisaat sejak papan dihidupkan. Digunakan untuk pemasaan tanpa membekukan program.

nyahlantun
: Teknik mengabaikan perubahan yang terlalu cepat (contohnya dalam 50 ms) supaya satu tekanan butang dikira sekali sahaja.

Nyahpepijat
: Proses mencari dan membetulkan punca kesilapan (pepijat) dalam program atau litar.

perintang tarik-naik
: Perintang yang menarik pin ke HIGH apabila tiada apa yang menekannya ke LOW. ESP32 dan Uno ada perintang dalaman yang dihidupkan dengan INPUT_PULLUP.

Rintangan
: Sesuatu yang menentang aliran arus. Unit: ohm (Ω).

semakan kod
: Proses membaca kod dengan teliti untuk mencari kesilapan dan memastikan ia memenuhi keperluan, sebelum kod digunakan.

sistem terbenam
: Sistem komputer kecil yang dibina di dalam sesuatu alat untuk satu tugas khusus, contohnya pengawal mesin basuh.

sketch
: Nama untuk program Arduino. Fail sketch berakhir dengan .ino.

spesifikasi
: Dokumen yang menerangkan dengan tepat apa yang sistem mesti buat, kekangannya, dan bagaimana ia akan diuji.

Voltan
: Daya yang menolak cas elektrik melalui litar. Unit: volt (V).

voltan hadapan
: Voltan yang jatuh merentasi LED apabila ia menyala. Merah lebih kurang 2.0 V, biru dan putih lebih kurang 3.0 V.