



UNIVERSITI  
TEKNOLOGI  
PETRONAS

# LAB WEEK 11

SEP 2025 SEMESTER

**'AFFAN NAJIY BIN RUSDI**

22010453

BACHELOR OF COMPUTER SCIENCE

INTERNET OF THINGS

TFB2093

# Contents

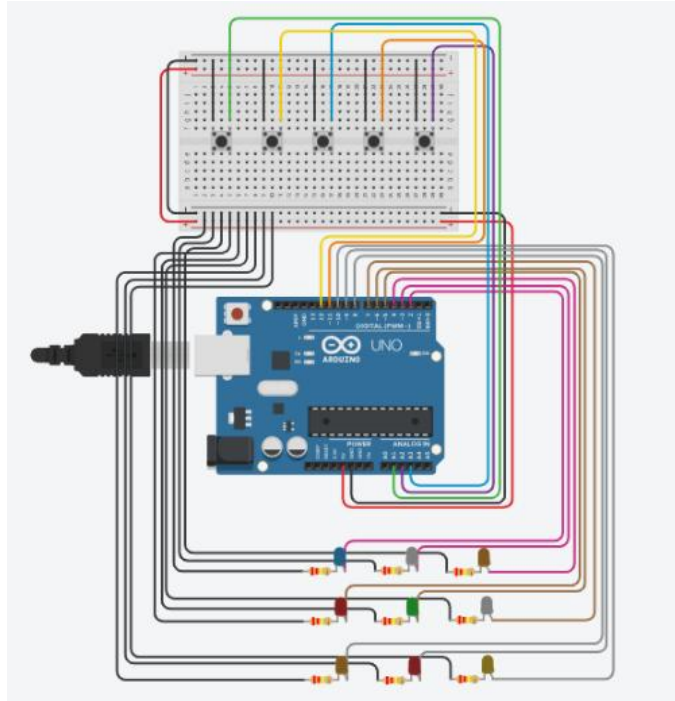
Code and Design ..... 3

## Code and Design

Activity 4 Link: <https://www.tinkercad.com/things/fopX6BWCx7X-lab8w11-4>

Activity 5 Link: <https://www.tinkercad.com/things/69T046f56Jp-lab8w11-5>

### Activity 4 and 5



## Code

### Activity 4

```
//Activity 4

const int ledPins[3][3] = {
  {2, 3, 4},    // Row 0
  {5, 6, 7},    // Row 1
  {8, 9, 10}    // Row 2
};

const int BTN_UP    = 11;
const int BTN_DOWN  = 12;
const int BTN_LEFT  = A1;
const int BTN_RIGHT = A2;
const int BTN_OK    = A3;    // belum guna lagi, untuk Activity 5 nanti

int playerRow = 1;    // start kat tengah
int playerCol = 1;    // start kat tengah

void setup() {
  // Set semua LED pins jadi OUTPUT
  for(int r = 0; r < 3; r++) {
    for(int c = 0; c < 3; c++) {
      pinMode(ledPins[r][c], OUTPUT);
    }
  }

  // Set button pins dengan internal pull-up
  pinMode(BTN_UP,    INPUT_PULLUP);
  pinMode(BTN_DOWN,  INPUT_PULLUP);
  pinMode(BTN_LEFT,  INPUT_PULLUP);
  pinMode(BTN_RIGHT, INPUT_PULLUP);
  pinMode(BTN_OK,    INPUT_PULLUP);

  // Nyalakan pixel tengah dulu
  digitalWrite(ledPins[playerRow][playerCol], HIGH);
}

void clearAll() {
  for(int r = 0; r < 3; r++) {
    for(int c = 0; c < 3; c++) {
      digitalWrite(ledPins[r][c], LOW);
    }
  }
}

void drawPlayer() {
  clearAll();
  digitalWrite(ledPins[playerRow][playerCol], HIGH);
}

void loop() {
  // Baca button dan gerak (hanya gerak kalau tak keluar grid)
  if (digitalRead(BTN_UP)    == LOW && playerRow > 0) playerRow--;
  if (digitalRead(BTN_DOWN)  == LOW && playerRow < 2) playerRow++;
  if (digitalRead(BTN_LEFT)  == LOW && playerCol > 0) playerCol--;
  if (digitalRead(BTN_RIGHT) == LOW && playerCol < 2) playerCol++;

  drawPlayer();

  delay(150);    // ni penting! debounce + buat gerak tak terlalu laju
}
```

**Activity 5**

```
//Activity 5

const int ledPins[3][3] = {
  {2, 3, 4},
  {5, 6, 7},
  {8, 9, 10}
};

const int BTN_UP    = 11;
const int BTN_DOWN  = 12;
const int BTN_LEFT  = A1;
const int BTN_RIGHT = A2;
const int BTN_OK    = A3;

int score = 0; // ← STUDENT TASK: Score counter
unsigned long targetTime;
bool targetOn = false;
int targetRow, targetCol;

void clearAll() {
  for (int r = 0; r < 3; r++) {
    for (int c = 0; c < 3; c++) {
      digitalWrite(ledPins[r][c], LOW);
    }
  }
}

void showPattern(const byte pattern[3][3]) {
  clearAll();
  for (int r = 0; r < 3; r++) {
    for (int c = 0; c < 3; c++) {
      digitalWrite(ledPins[r][c], pattern[r][c]);
    }
  }
}

void newTarget() {
  targetRow = random(0, 3);
  targetCol = random(0, 3);
  clearAll();
  digitalWrite(ledPins[targetRow][targetCol], HIGH);
  targetTime = millis();
  targetOn = true;
}

// ← STUDENT TASK: WIN pattern (smiley)
void winPattern() {
  byte win[3][3] = {{1,0,1},{0,1,0},{1,1,1}};
  showPattern(win);
  delay(1000);
}

// ← STUDENT TASK: LOSE pattern (cross)
void losePattern() {
  byte lose[3][3] = {{1,0,1},{0,1,0},{1,0,1}};
  showPattern(lose);
  delay(1000);
}

void setup() {
  Serial.begin(9600);
  randomSeed(analogRead(A5)); // Random seed for random()

  // LED pins
```

```

for (int r = 0; r < 3; r++) {
  for (int c = 0; c < 3; c++) {
    pinMode(ledPins[r][c], OUTPUT);
  }
}

// Buttons
pinMode(BTN_UP, INPUT_PULLUP);
pinMode(BTN_DOWN, INPUT_PULLUP);
pinMode(BTN_LEFT, INPUT_PULLUP);
pinMode(BTN_RIGHT, INPUT_PULLUP);
pinMode(BTN_OK, INPUT_PULLUP);

Serial.println("Reaction Game: Press CENTER on the lit LED!");
newTarget(); // Start first target
}

void loop() {
  if (targetOn) {
    // CENTER press: WIN
    if (digitalRead(BTN_OK) == LOW) {
      unsigned long reaction = millis() - targetTime;
      Serial.print("WIN! Reaction time: ");
      Serial.print(reaction);
      Serial.println(" ms");
      score++; // ← STUDENT TASK: Increase score
      Serial.print("Score: ");
      Serial.println(score);
      winPattern();
      delay(500); // Short pause
      newTarget();
    }

    // Timeout: LOSE
    if (millis() - targetTime > 2000) { // ← STUDENT TASK: Reduced time (2000 ms = harder)
      Serial.println("Too slow! LOSE");
      score = 0; // ← STUDENT TASK: Reset score on lose
      Serial.print("Score reset to: ");
      Serial.println(score);
      losePattern();
      delay(500); // Short pause
      newTarget();
    }
  }
}

```

