

# **LAPORAN PRAKTIKUM**

**PEMROGRAMAN BERORIENTASI OBJEK (PBO) – [TUGAS 4]**



**Disusun Oleh**  
Reza Chairul Manam  
120140086

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI SUMATERA  
2025**

# 1 Soal Quiz — Simple Minesweeper

## 1.1 Input Soal

Kamu diminta untuk membuat sebuah game Minesweeper sederhana dengan ukuran papan 3 baris x 3 kolom. Mekanisme permainan sangat sederhana: Akan ada 1 bom yang ditempatkan secara acak di salah satu kotak pada papan. Posisi bom ditentukan dari koordinat baris dan kolom secara acak. Gunakan simbol berikut untuk menampilkan kondisi papan: ? → Menandakan kotak belum dibuka. O → Menandakan kotak sudah dibuka dan tidak berisi bom. X → Menandakan kotak berisi bom dan telah dibuka. Saat pemain memilih kotak: Jika kotak tersebut tidak berisi bom, tampilkan O dan lanjutkan permainan. Jika kotak tersebut berisi bom, tampilkan X dan berikan pesan seperti: "Boom! Kamu kalah!" Jika pemain berhasil membuka semua kotak tanpa mengenai bom, tampilkan pesan seperti: "Selamat! Kamu menang!"

## 1.2 Source Code

```
# Quiz:
import random

class Cell:
    def __init__(self):
        self.is_bomb = False
        self.revealed = False

    def reveal(self):
        self.revealed = True

    def display(self):
        if self.revealed:
            return "X" if self.is_bomb else "O"
        else:
            return "?"

class Minesweeper:
    def __init__(self, rows=3, cols=3):
        self.rows = rows
        self.cols = cols
        self.board = [[Cell() for _ in range(cols)] for _
                        in range(rows)]
        self.bomb_row = random.randint(0, rows - 1)
```

```

        self.bomb_col = random.randint(0, cols - 1)
        self.board[self.bomb_row][self.bomb_col].is_bomb =
            True
        self.game_over = False

    def print_board(self):
        print("=" * 40)
        for row in self.board:
            print(" ".join(cell.display() for cell in row))
        print("=" * 40)

    def play_turn(self, row, col):
        if self.board[row][col].revealed:
            print("Kotak ini sudah dibuka.")
            return

        self.board[row][col].reveal()
        if self.board[row][col].is_bomb:
            self.game_over = True
            self.print_board()
            print("Boom! Kamu kalah!")
        else:
            print("Tidak ada bom di sini. Lanjutkan!")
            if self.check_win():
                self.print_board()
                print("Selamat! Kamu menang!")

    def check_win(self):
        for row in self.board:
            for cell in row:
                if not cell.is_bomb and not cell.revealed:
                    return False
        self.game_over = True
        return True

    def run(self):
        while not self.game_over:
            self.print_board()
            try:
                row = int(input("Masukkan baris (0-2): "))
                col = int(input("Masukkan kolom (0-2): "))
                if 0 <= row < self.rows and 0 <= col < self
                    .cols:

```

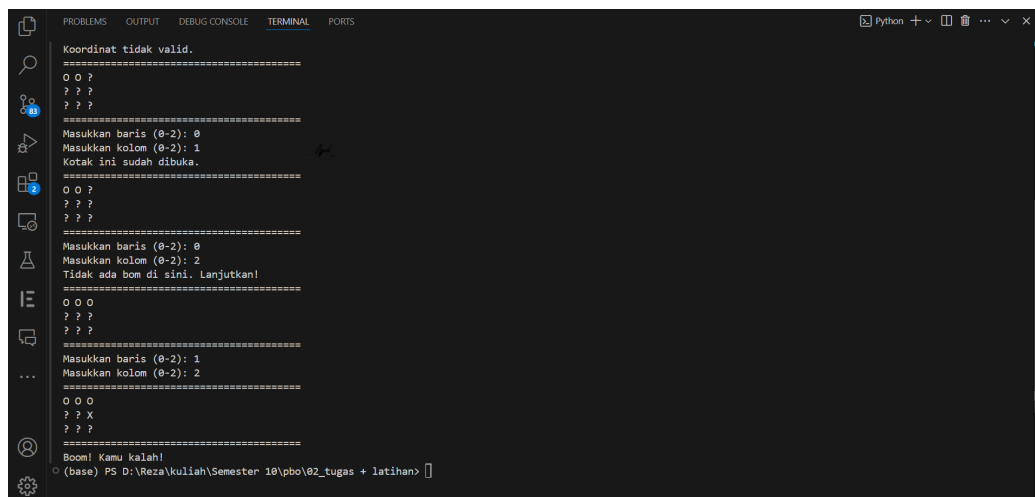
```

        self.play_turn(row, col)
    else:
        print("Koordinat tidak valid.")
except ValueError:
    print("Masukkan angka yang valid.")

# Jalankan game
game = Minesweeper()
game.run()

```

### 1.3 Output Hasil



```

Koordinat tidak valid.
=====
O O ?
? ? ?
? ? ?
=====
Masukkan baris (0-2): 0
Masukkan kolom (0-2): 1
Kotak ini sudah dibuka.
=====
O O ?
? ? ?
? ? ?
=====
Masukkan baris (0-2): 0
Masukkan kolom (0-2): 2
Tidak ada bom di sini. Lanjutkan!
=====
O O O
? ? ?
? ? ?
=====
Masukkan baris (0-2): 1
Masukkan kolom (0-2): 2
=====
O O O
? ? X
? ? ?
=====
Boom! Kamu kalah!
(base) PS D:\Reza\kuliah\Semester 10\pbo\02_tugas + latihan>

```

### 1.4 Penjelasan

Program ini merupakan implementasi sederhana dari permainan Minesweeper menggunakan bahasa pemrograman Python. Papan permainan berukuran 3x3, dan hanya terdapat 1 buah bom yang dipasang secara acak di salah satu sel. Pemain akan diminta untuk memilih koordinat (baris dan kolom) untuk membuka kotak. Jika pemain membuka kotak yang berisi bom, permainan berakhir. Jika semua kotak yang tidak berisi bom berhasil dibuka, maka pemain menang.

## 2 Soal Nomor 2 — Manajemen Daftar Tugas (To-Do List)

### 2.1 Input Soal

Pada kuis kali ini, kamu diminta untuk membuat game Hangman sederhana, yaitu permainan tebak kata di mana pemain menebak huruf demi huruf. Untuk gambar Hangman dan daftar kata yang akan ditebak: Sudah disiapkan oleh penyusun kuis, dan boleh langsung digunakan. Namun, jika kamu ingin berkreasi sendiri dengan gambar atau daftar kata baru, itu boleh juga. Semua terserah kamu! Setiap kali game dimainkan, kata yang akan ditebak harus dipilih secara acak dari daftar yang tersedia. Terakhir, kamu perlu menampilkan hasil akhir permainan, apakah: Pemain berhasil menebak kata dengan benar, atau Pemain gagal menebak kata dengan benar sampai kesempatan habis.

### 2.2 Source Code

```
# Quiz 2:
# File: 120140086_Quiz_2.py
# Author: Reza Chairul Manam
# NIM: 120140086
# Class: Praktikum PBO-RC
#
=====

# Quiz 2:
import random

class Hangman:
    def __init__(self, word_list, attempts=6):
        self.word_list = word_list
        self.word = random.choice(self.word_list)
        self.attempts_left = attempts
        self.guessed_letters = []
        self.correct_letters = ['_' for _ in self.word]

    def display_hangman(self):
        stages = [
            """
            -----
            |       |
            O       |
            """
```









tebakan akan ditampilkan hasilnya, termasuk gambar visual sederhana dari hangman yang berubah sesuai kesalahan. Program ini juga menunjukkan penggunaan kondisi, perulangan, dan manipulasi string dalam Python.

**Repositori GitHub Tugas:** Berikut adalah tautan menuju repositori GitHub yang berisi seluruh kode sumber tugas praktikum:

<https://github.com/rezachairul/Latihan-PBO-2025/tree/main/Praktikum>

**Dokumen Laporan di Overleaf:** Untuk melihat laporan ini secara langsung di Overleaf, gunakan tautan berikut:

<https://www.overleaf.com/read/dnyxfptcxtsj#44dd44>