



# **Laporan Program Python Praktikum Lab – Class dan Turtle**

## **Mata Kuliah:**

Pemrograman Berorientasi Objek

## **Oleh:**

Gerry Moeis Mahardika Dwi Putra

23091397164

2023E

<https://github.com/gerrymoeis/pbo-2>



**PROGRAM STUDI MANAJEMEN INFORMATIKA FAKULTAS VOKASI**

**UNIVERSITAS NEGERI SURABAYA**

**2024**



### Laporan Tugas 1: Membuat Game Uler-uleran

Langkah	Praktikum
1.	<pre>import pygame, sys, random, time  pygame.init()</pre> <p>Disini saya import modul pygame yang telah diinstall, juga modul bawaan seperti sys, random, dan time. Lalu lakukan init. Untuk menginisiasi semua komponen atau fungsi dalam pygame.</p> <pre>class GameObject:     def __init__(self, color, position, size):         self.color = color         self.position = position         self.size = size</pre> <p>Selanjutnya, saya buat class GameObject. Sebagai blueprint parent pembuatan object maupun class inheritance lainnya.</p> <p>Atribut yang ditambahkan adalah parameter umum.</p> <pre>def draw(self, game_window):     pygame.draw.rect(game_window, self.color, pygame.Rect(self.position[0], self.position[1], self.size, self.size))</pre> <p>Dalam class GameObject kita buat metode utama yaitu draw. Yang mana bertujuan untuk menggambar objek yang diharapkan sesuai atributnya.</p>

2.

```
class Snake(GameObject):
    def __init__(self, color, position, size, speed=10):
        super().__init__(color, position, size)
        self.body = [list(position), [position[0] - size, position[1]],
[position[0] - size, position[1]]]
        self.speed = speed
        self.direction = 'RIGHT'
        self.change_to = self.direction
```

Sekarang kita buat Inheritance dari GameObject yaitu Snake. Objek Snake ini akan memiliki atribut dari GameObject juga tambahan beberapa atribut yang spesifik untuk kebutuhannya.

```
def change_direction(self, direction):
    if direction == 'UP' and self.direction != 'DOWN':
        self.change_to = 'UP'
    if direction == 'DOWN' and self.direction != 'UP':
        self.change_to = 'DOWN'
    if direction == 'LEFT' and self.direction != 'RIGHT':
        self.change_to = 'LEFT'
    if direction == 'RIGHT' and self.direction != 'LEFT':
        self.change_to = 'RIGHT'
```

Lalu method selanjutnya pada class Snake adalah change\_direction, yang bertujuan menyimpan data variabel arah untuk digunakan nantinya.

```
def move(self):
    self.direction = self.change_to
    if self.direction == 'UP':
        self.position[1] -= self.speed
    if self.direction == 'DOWN':
        self.position[1] += self.speed
    if self.direction == 'LEFT':
        self.position[0] -= self.speed
    if self.direction == 'RIGHT':
        self.position[0] += self.speed
    self.body.insert(0, list(self.position))
```

Nah berikutnya ada method move. Yang mana mengatur pergerakan Snake nantinya. Arah dan pergerakan dari kode sebelumnya di terapkan pada method ini, lalu ditambah kecepatan sesuai arah gerakan.

```
def shrink(self):
    self.body.pop()
```

Method shrink disini bertujuan untuk menghapus ekor snake, karena nanti saat di



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,  
RISET, DAN TEKNOLOGI  
**UNIVERSITAS NEGERI SURABAYA**

Kampus Unesa 1, jalan Ketintang Surabaya 60231

Laman: <https://vokasi.unesa.ac.id/> E-mail: [vokasi@unesa.ac.id](mailto:vokasi@unesa.ac.id)

gambar, jika ekor tidak dihapus, maka panjang snake akan terus menambah.

```
def draw(self, game_window):  
    for pos in self.body:  
        pygame.draw.rect(game_window, self.color, pygame.Rect(pos[0],  
pos[1], self.size, self.size))
```

Nah untuk method draw pada blueprint GameObject sebelumnya, disini kita mengubahnya sesuai kebutuhan kita, yaitu dengan looping yang bertujuan menggambar masing-masing bagian Snake.

```
def check_collision(self, frame_size_x, frame_size_y):  
    if self.position[0] < 0 or self.position[0] > frame_size_x - self.size  
or self.position[1] < 0 or self.position[1] > frame_size_y - self.size:  
        return True  
  
    for block in self.body[1:]:  
        if self.position[0] == block[0] and self.position[1] == block[1]:  
            return True  
  
    return False
```

Kemudian penting kita tambahkan method check\_collision untuk melakukan cek bilamana Snake menabrak atau bersinggungan dengan GameObject lain.



3.	<pre>class Apple(GameObject):     def __init__(self, frame_size_x, frame_size_y, color):         self.position = [random.randrange(1, (frame_size_x // 10)) * 10, random.randrange(1, (frame_size_y // 10)) * 10]         super().__init__(color, self.position, 10)</pre> <p>Sekarang kita buat class Apple yang juga inheritance dari GameObject. Untuk inisiasi atribut. Disini kita mengkustomisasi atribut position.</p> <pre>def respawn(self, frame_size_x, frame_size_y):     self.position = [random.randrange(1, (frame_size_x // 10)) * 10, random.randrange(1, (frame_size_y // 10)) * 10]</pre> <p>Nah method yang kita tambahkan pada class Apple ini ialah respawn. Bertujuan untuk memunculkan apple kembali, nanti untuk mechanicnya akan dijelaskan pada langkah-langkah berikutnya.</p>
4.	<pre>class Game:     def __init__(self, apple_amount):         self.frame_size_x = 720         self.frame_size_y = 480         self.game_window = pygame.display.set_mode((self.frame_size_x, self.frame_size_y))         pygame.display.set_caption('Aowkwk')          self.fps_controller = pygame.time.Clock()         self.snake = Snake(pygame.Color(0, 255, 0), [100, 50], 10)         self.score = 0          self.apples = []         for _ in range(apple_amount):             self.apples.append(Apple(self.frame_size_x, self.frame_size_y, pygame.Color(random.randint(0, 255), random.randint(0, 255), random.randint(0, 255))))          self.arena_position = (50, 50)         self.arena_size = (self.frame_size_x - 100, self.frame_size_y - 100)</pre> <p>Oke, class terakhir yang kita butuhkan adalah class Game. Objek ini nantinya akan berfungsi untuk mengatur keseluruhan mechanic game termasuk layar tampilan permainannya.</p> <p>Terdapat banyak atribut yang didefinisikan di class ini. Mulai dari ukuran layar, window, snake, apple, hingga arena.</p>



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,  
RISET, DAN TEKNOLOGI  
**UNIVERSITAS NEGERI SURABAYA**

Kampus Unesa 1, jalan Ketintang Surabaya 60231

Laman: <https://vokasi.unesa.ac.id/> E-mail: [vokasi@unesa.ac.id](mailto:vokasi@unesa.ac.id)

```
def draw_arena(self):  
    pygame.draw.rect(self.game_window, pygame.Color(0, 0, 0),  
pygame.Rect(self.arena_position[0], self.arena_position[1], self.arena_size[0],  
self.arena_size[1]), 2)
```

Pertama kita buat method draw\_arena untuk menggambar kotak arena nantinya.

```
def game_over(self):  
    my_font = pygame.font.SysFont('Times New Roman', 90)  
    game_over_surface = my_font.render('YOU DIED', True, pygame.Color(255,  
0, 0))  
  
    game_over_rect = game_over_surface.get_rect()  
    game_over_rect.midtop = (self.frame_size_x / 2, self.frame_size_y / 4)  
    self.game_window.fill(pygame.Color(0, 0, 0))  
    self.game_window.blit(game_over_surface, game_over_rect)  
  
    pygame.display.flip()  
    time.sleep(3)  
    pygame.quit()  
    sys.exit()
```

Lalu method game\_over, yang mengatur bilamana Snake menabrak dinding atau menabrak dirinya sendiri maka ditampilkan Game Over dan di exit.

```
def show_score(self):  
    score_font = pygame.font.SysFont('Times New Roman', 20)  
    score_surface = score_font.render('Score : ' + str(self.score), True,  
pygame.Color(0, 0, 0))  
    score_rect = score_surface.get_rect()  
    score_rect.midtop = (72, 15)  
    self.game_window.blit(score_surface, score_rect)
```

Selanjutnya method show\_score untuk menampilkan jumlah score yang diperoleh. Disini ditetapkan font, warna, dan format teks yang akan ditampilkan.



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,  
RISET, DAN TEKNOLOGI  
**UNIVERSITAS NEGERI SURABAYA**

Kampus Unesa 1, jalan Ketintang Surabaya 60231

Laman: <https://vokasi.unesa.ac.id/> E-mail: [vokasi@unesa.ac.id](mailto:vokasi@unesa.ac.id)

```
def run(self):
    while True:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                sys.exit()
            elif event.type == pygame.KEYDOWN:
                if event.key == pygame.K_UP:
                    self.snake.change_direction('UP')
                if event.key == pygame.K_DOWN:
                    self.snake.change_direction('DOWN')
                if event.key == pygame.K_LEFT:
                    self.snake.change_direction('LEFT')
                if event.key == pygame.K_RIGHT:
                    self.snake.change_direction('RIGHT')

                if event.key == pygame.K_ESCAPE:
                    pygame.event.post(pygame.event.Event(pygame.QUIT))

        self.snake.move()

        if self.snake.position in map(lambda apple: apple.position,
self.apples):
            eated_apple = [apple for apple in self.apples if apple.position
== self.snake.position][0]
            eated_apple.respawn(self.arena_size[0], self.arena_size[1])

            self.score += 1
        else:
            self.snake.shrink()

        self.game_window.fill(pygame.Color(255, 255, 255))

        self.draw_arena()
        self.snake.draw(self.game_window)
        self.show_score()

        for apple in self.apples:
            apple.draw(self.game_window)

        if self.snake.check_collision(self.frame_size_x, self.arena_size[1]
+ self.arena_position[1]):
            self.game_over()

        pygame.display.update()
        self.fps_controller.tick(10)
```

	<p>Terakhir method paling penting yaitu run. Method yang mengatur keseluruhan jalannya permainan dan mechanicnya.</p> <p>Untuk penjelasan masing-masing kode akan diterangkan lebih detail di bawah:</p> <pre> while True:     for event in pygame.event.get():         if event.type == pygame.QUIT:             pygame.quit()             sys.exit()         elif event.type == pygame.KEYDOWN:             if event.key == pygame.K_UP:                 self.snake.change_direction('UP')             if event.key == pygame.K_DOWN:                 self.snake.change_direction('DOWN')             if event.key == pygame.K_LEFT:                 self.snake.change_direction('LEFT')             if event.key == pygame.K_RIGHT:                 self.snake.change_direction('RIGHT')              if event.key == pygame.K_ESCAPE:                 pygame.event.post(pygame.event.Event(pygame.QUIT))      self.snake.move() </pre> <p>Tujuan kode di atas untuk menggerakkan snake berdasarkan input panah pada keyboard.</p> <pre> if self.snake.position in map(lambda apple: apple.position, self.apples):     eated_apple = [apple for apple in self.apples if apple.position == self.snake.position][0]     eated_apple.respawn(self.arena_size[0], self.arena_size[1])      self.score += 1 else:     self.snake.shrink() </pre> <p>Lalu disini kita buat kondisi yang mana untuk mengecek bilamana Snake memakan apple (posisi keduanya bersinggungan / collide). Maka score ditambah dan apple di respawn.</p> <p>Tetapi bilamana snake sedang tidak memakan apple. Maka snake di shrink agar ukuran snake tetap sama.</p>
--	--





KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,  
RISET, DAN TEKNOLOGI  
**UNIVERSITAS NEGERI SURABAYA**

Kampus Unesa 1, jalan Ketintang Surabaya 60231

Laman: <https://vokasi.unesa.ac.id/> E-mail: [vokasi@unesa.ac.id](mailto:vokasi@unesa.ac.id)

	<pre>self.game_window.fill(pygame.Color(255, 255, 255))  self.draw_arena() self.snake.draw(self.game_window) self.show_score()  for apple in self.apples:     apple.draw(self.game_window)</pre> <p>Kode di atas adalah kumpulan kode untuk menampilkan UI. Seperti menggambar arena, snake, apple, dsb.</p> <pre>if self.snake.check_collision(self.frame_size_x, self.arena_size[1] + self.arena_position[1]):     self.game_over()</pre> <p>Nah disini kita cek terlebih dahulu soal collision snake. Bilamana snake menabrak dinding arena maka tampilkan pesan Game Over.</p> <pre>pygame.display.update() self.fps_controller.tick(10)</pre> <p>Terakhir, kode di atas bertujuan untuk mengatur frame pada window pygame, disitu window akan di update dan fps ditentukan.</p>



## Laporan Tugas 2: Penambahan Apple

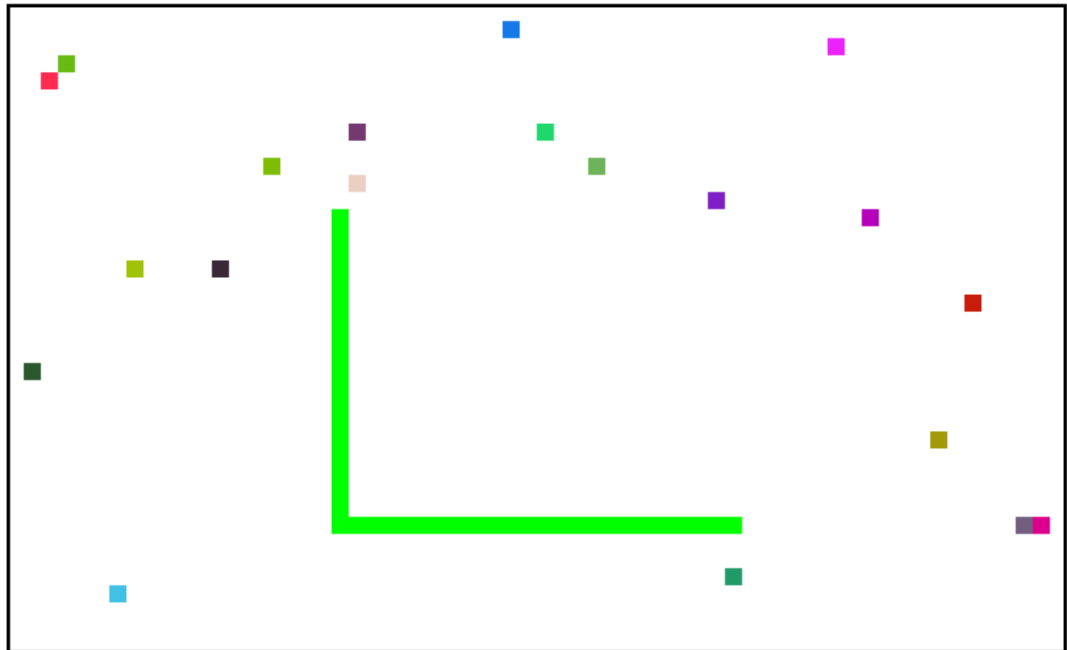
Langkah	Praktikum
1.	<pre>self.apples = [] for _ in range(apple_amount):     self.apples.append(Apple(self.frame_size_x, self.frame_size_y, pygame.Color(random.randint(0, 255), random.randint(0, 255), random.randint(0, 255))))</pre> <p>Untuk kode ini sebelumnya telah dilampirkan. Tetapi akan saya jelaskan lebih lanjut.</p> <p>Jadi disini saya membuat list apples sebagai atribut class Game. Lalu saya loop dari parameter yang saya tambahkan pada class Game, untuk menambah Objek Apple sejumlah apple_amount.</p> <pre>if self.snake.position in map(lambda apple: apple.position, self.apples):     eated_apple = [apple for apple in self.apples if apple.position == self.snake.position][0]     eated_apple.respawn(self.arena_size[0], self.arena_size[1])</pre> <p>Berikutnya untuk mechanic snake memakan apple, perlu kita konfigurasi agar hanya apple yang dimakan itu bisa respawn.</p> <p>Teknisnya disini saya menggunakan fungsi map, lalu lambda function. Lalu kita tampung dulu posisi eated_apple dalam variable. Lalu kita respawn eated_apple tersebut.</p> <pre>for apple in self.apples:     apple.draw(self.game_window)</pre> <p>Terakhir kita gambar masing-masing apple pada layar window pygame menggunakan looping.</p>



**Output:  
Game:**

Aowkwk

Score : 39



**Game Over:**

Aowkwk





KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,  
RISET, DAN TEKNOLOGI  
**UNIVERSITAS NEGERI SURABAYA**  
Kampus Unesa 1, jalan Ketintang Surabaya 60231  
Laman: <https://vokasi.unesa.ac.id/> E-mail: [vokasi@unesa.ac.id](mailto:vokasi@unesa.ac.id)

---

***--#BeraksiBerprestasiBersinergi--***