# GRAFIKA KOMPUTER

Dosen Pengampu : CUCUN VERY ANGKOSO M.T S.T



Disusun Oleh :

**TORY FIANATA**

210411100170

# TEKNIK INFORMATIKA FAKULTAS TEKNIK UNIVERSITAS TRUNOJOYO MADURA

**2023**

**Laporan: Implementasi Aplikasi "Bouncing Balls" menggunakan Pygame**

**1. Pendahuluan**

Aplikasi "Bouncing Balls" dibuat menggunakan library Pygame untuk mensimulasikan pergerakan bola-bola yang memantul di layar. Laporan ini akan menjelaskan detail teknis implementasi dari aplikasi ini.

**2. Deskripsi Teknis**

**2.1. Inisialisasi Pygame**

import pygame

import random

# Inisialisasi Pygame

pygame.init()

# Ukuran layar

screen\_width = 800

screen\_height = 600

screen = pygame.display.set\_mode((screen\_width, screen\_height))

pygame.display.set\_caption("Bouncing Balls")

* **pygame.init():** Memulai Pygame dan menginisialisasi semua modul yang diperlukan.
* **screen\_width** dan **screen\_height:** Menentukan ukuran jendela aplikasi.
* **pygame.display.set\_mode():** Membuat jendela layar dengan ukuran yang telah ditentukan sebelumnya.

**2.2. Kelas Ball**

# Kelas Bola

class Ball:

    def \_\_init\_\_(self, x, y, radius, color):

        self.x = x

        self.y = y

        self.radius = radius

        self.color = color

        self.dx = random.choice([-5, 5])

        self.dy = random.choice([-5, 5])

    def move(self):

        self.x += self.dx

        self.y += self.dy

        # Pantulan dinding

        if self.x - self.radius < 0 or self.x + self.radius > screen\_width:

            self.dx = -self.dx

        if self.y - self.radius < 0 or self.y + self.radius > screen\_height:

            self.dy = -self.dy

    def draw(self):

        pygame.draw.circle(screen, self.color, (self.x, self.y), self.radius)

* **Ball:** Kelas yang merepresentasikan objek bola dalam aplikasi.
* **init(self, x, y, radius, color):** Metode inisialisasi untuk mengatur posisi awal, ukuran (radius), warna, dan kecepatan awal (dx dan dy) bola.
* **move(self):** Metode untuk menggerakkan bola dengan mengubah posisi x dan y berdasarkan dx dan dy, serta menangani pantulan bola saat mencapai batas layar.
* **draw(self):** Metode untuk menggambar bola menggunakan fungsi pygame.draw.circle(), dengan posisi dan warna yang telah ditentukan.

# Main loop

running = True

purple\_ball\_mode = "bounce"

while running:

    for event in pygame.event.get():

        if event.type == pygame.QUIT:

            running = False

        if event.type == pygame.KEYDOWN:

            if event.key == pygame.K\_LEFT:

                purple\_ball.dx = -10

                purple\_ball.dy = 0

            if event.key == pygame.K\_RIGHT:

                purple\_ball.dx = 10

                purple\_ball.dy = 0

            if event.key == pygame.K\_UP:

                purple\_ball.dx = 0

                purple\_ball.dy = -10

            if event.key == pygame.K\_DOWN:

                purple\_ball.dx = 0

                purple\_ball.dy = 10

            if event.key == pygame.K\_SPACE:

                purple\_ball\_mode = "stop" if purple\_ball\_mode == "bounce" else "bounce"

    screen.fill(black)

    pygame.draw.rect(screen, white, (0, 0, screen\_width, screen\_height), 5)

    for ball in balls:

        ball.move()

        ball.draw()

    for i in range(len(balls)):

        for j in range(i + 1, len(balls)):

            if (balls[i].x - balls[j].x)\*\*2 + (balls[i].y - balls[j].y)\*\*2 < (balls[i].radius + balls[j].radius)\*\*2:

                balls[i].dx, balls[j].dx = balls[j].dx, balls[i].dx

                balls[i].dy, balls[j].dy = balls[j].dy, balls[i].dy

                if balls[i] == purple\_ball or balls[j] == purple\_ball:

                    hit\_count += 1

                if purple\_ball\_mode == "stop":

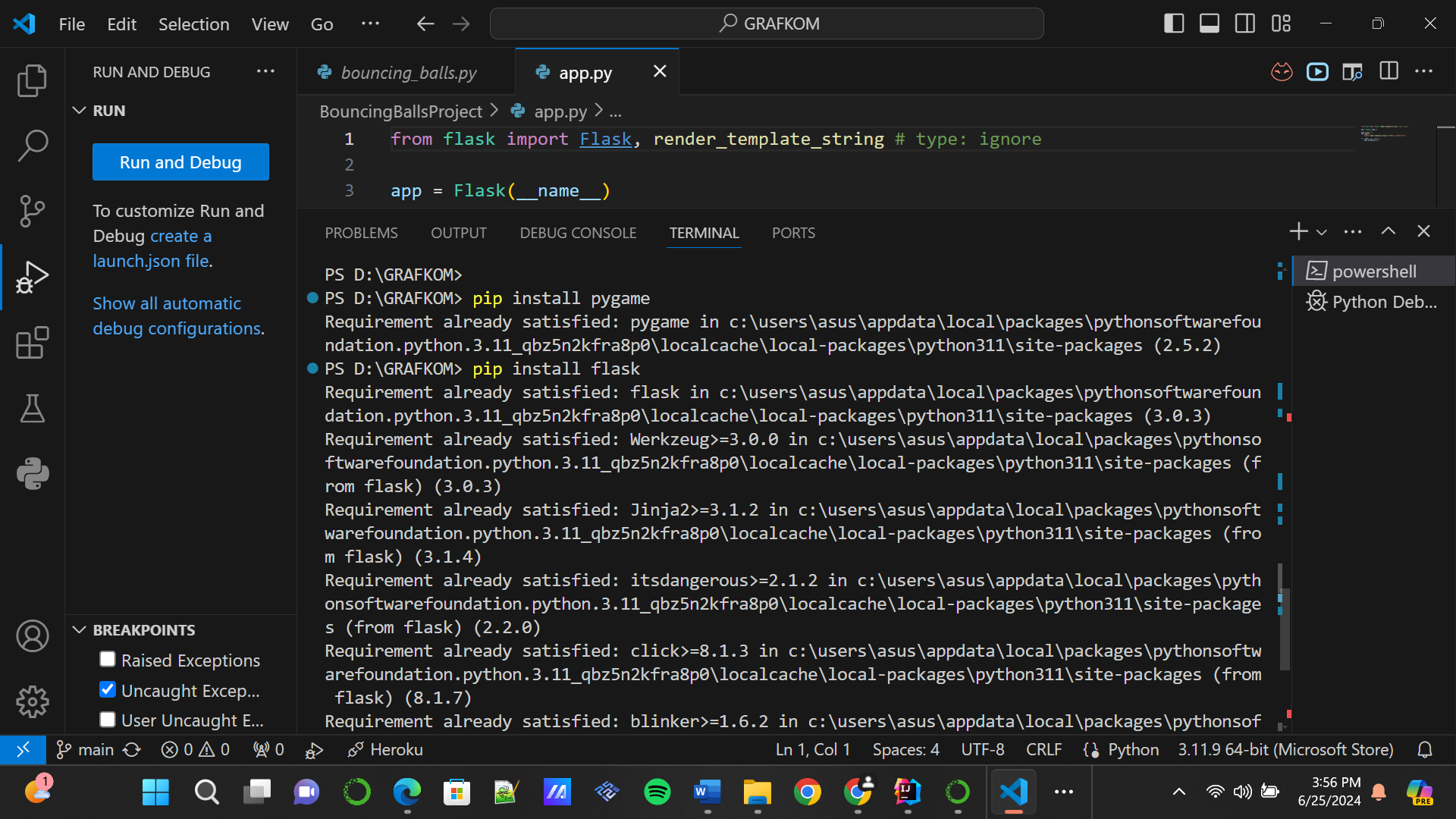
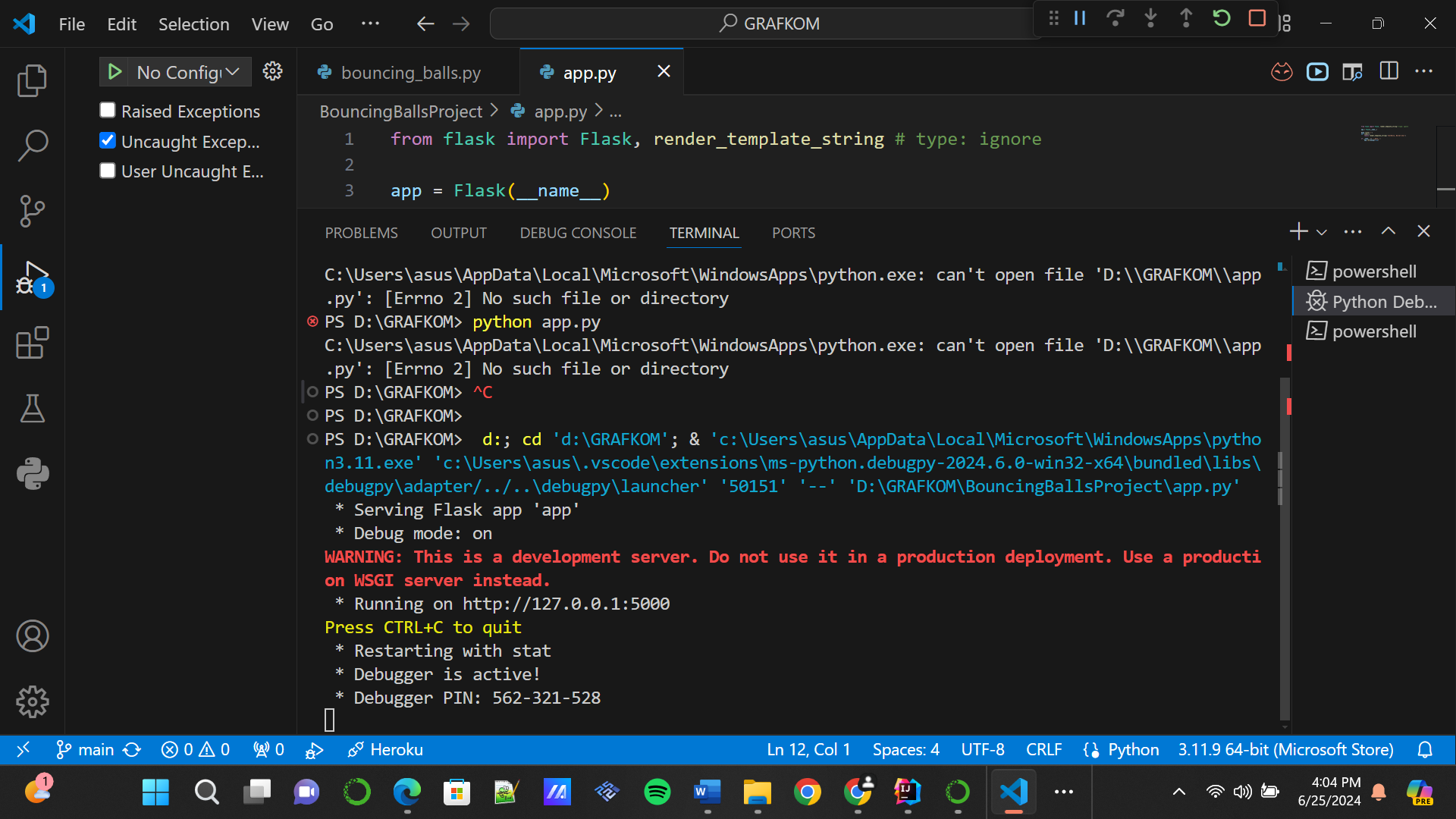
                    purple\_ball.dx, purple\_ball.dy = 0, 0

    display\_text(f'Score: {hit\_count}', 10, 10)

    pygame.display.flip()

    pygame.time.delay(20)

pygame.quit()

* **Main Loop:** Loop utama yang menjalankan aplikasi, mengendalikan pergerakan bola, menangani interaksi pengguna (input keyboard), serta menampilkan hasilnya pada layar.
* **pygame.event.get():** Mendapatkan semua event dari pygame seperti tombol keyboard yang ditekan atau menutup jendela.
* **Event handling:** Mengubah arah pergerakan bola ungu dan mengubah mode (bouncing atau stop) berdasarkan input dari keyboard.
* **Update layar:** Menggambar background, frame, semua bola, dan skor hit\_count pada setiap iterasi loop.
* Penginstalan Flask dan pygame
* Menjalankan flask dengan mengetik python app.py 

**3. Implementasi Tambahan**

* **Pantulan antar bola:** Logika untuk mendeteksi dan mengatur tabrakan antar bola, serta memantulkan bola-bola tersebut dari dinding.
* **Kontrol bola ungu:** Menggunakan keyboard untuk mengontrol pergerakan bola ungu dan mengubah mode pergerakan (bouncing atau stop).
* **Perhitungan skor:** Menghitung dan menampilkan skor hit\_count berdasarkan tabrakan bola ungu dengan bola-bola lainnya.

## 4. Pengunggahan ke GitHub

### 4.1. Persiapan Repositori

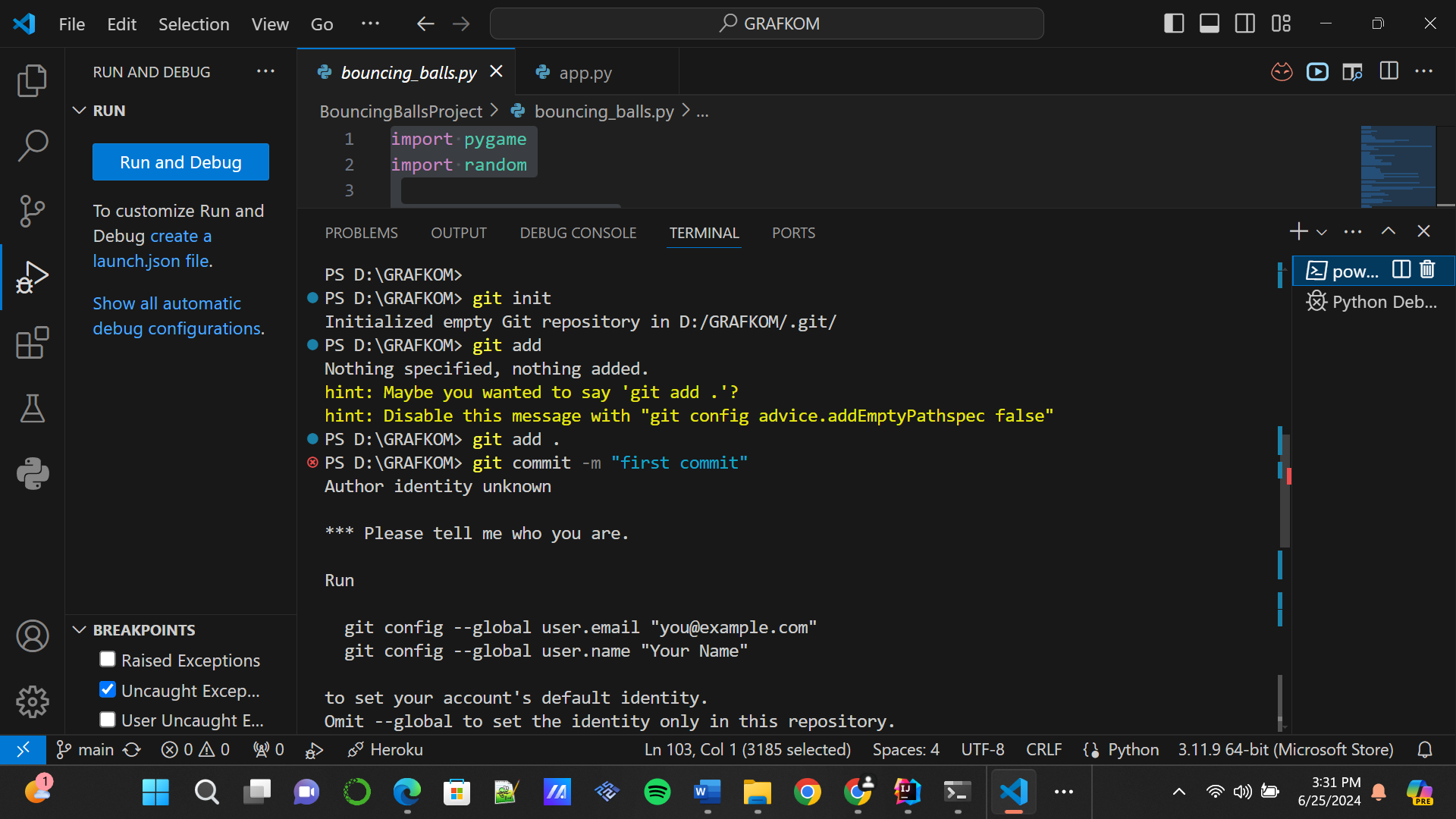
Sebelum mengunggah proyek Anda ke GitHub, pastikan untuk membuat repositori baru di GitHub dan menghubungkannya dengan proyek lokal Anda menggunakan Git.

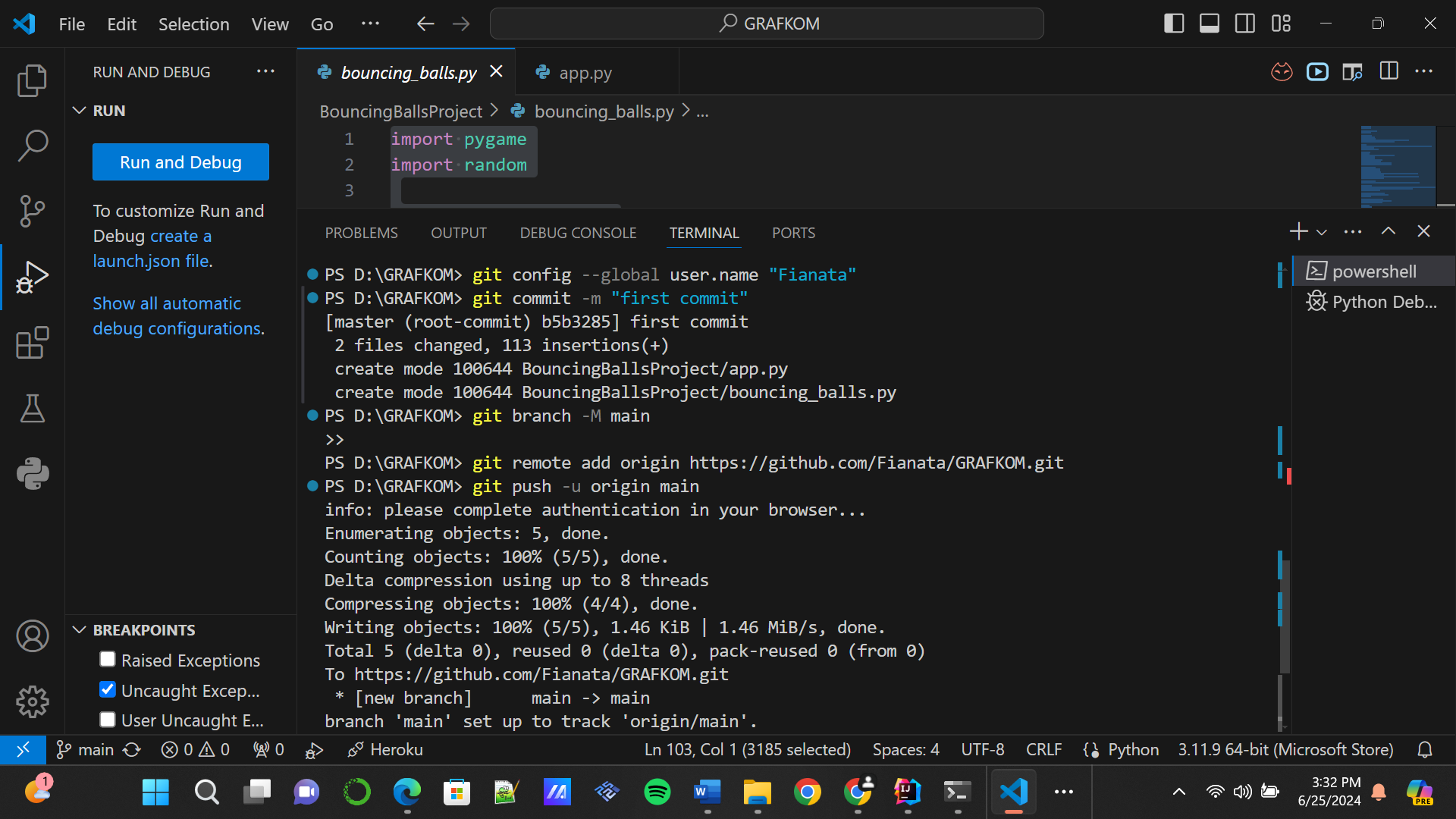
#### Langkah-langkahnya sebagai berikut:

1. **Buat Repositori di GitHub:**
   * Buka halaman GitHub dan buat repositori baru.
   * Beri nama repositori sesuai dengan proyek Anda dan tambahkan deskripsi jika diperlukan.
   * Jangan lupa untuk menyalakan opsi untuk membuat README file (jika belum ada) untuk dokumentasi.
2. **Inisialisasi Repositori Lokal:**
   * Buka terminal atau command prompt di direktori proyek Anda.
   * Jalankan perintah git init untuk menginisialisasi repositori lokal.
3. **Tambahkan File ke Git dan Buat Commit Pertama:**
   * Jalankan perintah git add . untuk menambahkan semua file proyek ke Git.
   * Jalankan perintah git commit -m "Initial commit" untuk membuat commit pertama dengan pesan yang jelas.
4. **Hubungkan Repositori Lokal dengan Repositori GitHub:**
   * Salin URL repositori GitHub yang telah Anda buat.
   * Jalankan perintah git remote add origin <URL\_repo\_GitHub> untuk menghubungkan repositori lokal dengan repositori GitHub.

### 4.2. Unggah Proyek ke GitHub

Setelah repositori terhubung, Anda dapat mengunggah proyek Anda dengan menjalankan perintah berikut:

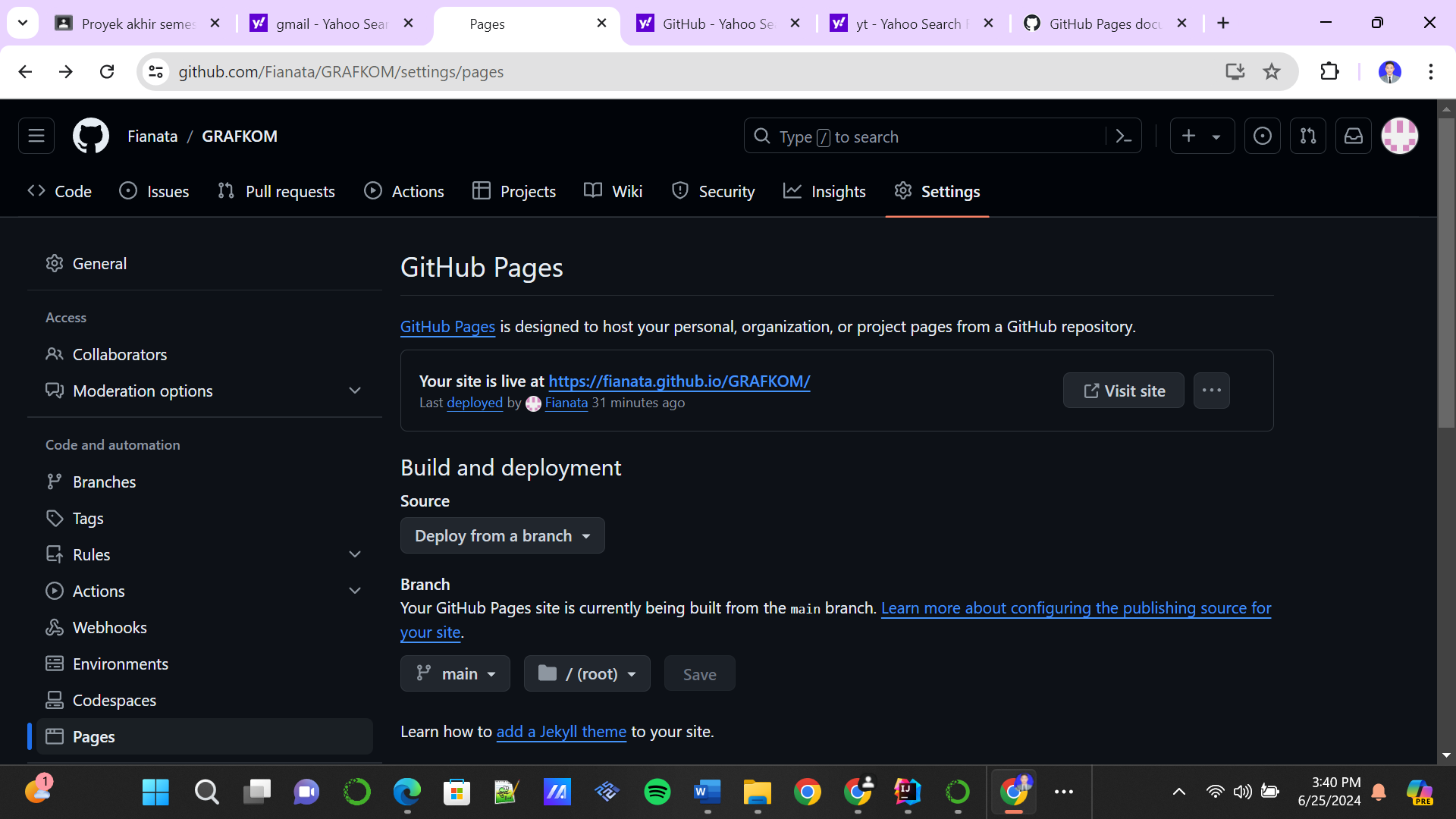




Pastikan untuk mengganti main dengan nama branch utama yang Anda gunakan (misalnya master atau main). Git akan meminta Anda untuk mengautentikasi dengan akun GitHub Anda.

### 4.3. Verifikasi dan Dokumentasi

* Pastikan proyek berhasil diunggah ke GitHub dengan membuka repositori Anda di browser.



* Jangan lupa untuk menambahkan README file yang menjelaskan proyek Anda, instalasi yang diperlukan, cara menjalankan aplikasi, dan detail lainnya yang relevan.

**5. Kesimpulan**

**5.1. Implementasi Aplikasi "Bouncing Balls" menggunakan Pygame**

Implementasi aplikasi "Bouncing Balls" menggunakan Pygame berhasil menunjukkan penerapan konsep dasar dalam pengembangan permainan sederhana berbasis grafis 2D. Melalui pembuatan kelas Ball dan penggunaan event handling, aplikasi ini mampu mensimulasikan pergerakan bola yang memantul di dalam area layar. Beberapa fitur utama yang berhasil diimplementasikan meliputi:

* **Kelas Ball:** Digunakan untuk merepresentasikan setiap bola dengan posisi, ukuran (radius), warna, dan kecepatan yang dapat diatur.
* **Pantulan Dinding:** Setiap bola dapat memantul ketika mencapai batas layar, yang diimplementasikan dengan mengubah arah bola (dx dan dy) ketika posisi bola mendekati batas layar.
* **Interaksi Pengguna:** Pengguna dapat mengendalikan bola ungu dengan keyboard untuk mengubah arah pergerakannya, serta mengaktifkan atau menonaktifkan mode pantulan bola ungu dengan tombol spasi.
* **Deteksi dan Reaksi Tabrakan Bola:** Aplikasi mampu mendeteksi tabrakan antara bola-bola dan meresponsnya dengan memantulkan bola-bola yang bersentuhan.

Dengan demikian, aplikasi ini tidak hanya mengeksplorasi kemampuan dasar Pygame dalam menangani grafika 2D, tetapi juga mengasah penggunaan event handling untuk interaksi pengguna yang dinamis.

**5.2. Pengunggahan Proyek ke GitHub**

Langkah selanjutnya setelah berhasil mengimplementasikan aplikasi adalah dengan mengunggahnya ke GitHub. Proses ini tidak hanya memungkinkan untuk berbagi kode dengan orang lain, tetapi juga membangun portofolio yang terorganisir dan dapat diakses secara publik. Beberapa langkah kunci dalam proses pengunggahan proyek ke GitHub meliputi:

* **Persiapan Repositori:** Pembuatan repositori baru di GitHub dan inisialisasi repositori lokal menggunakan Git.
* **Tambahkan dan Commit File:** Menambahkan semua file proyek ke Git dan membuat commit pertama dengan pesan yang jelas untuk dokumentasi.
* **Hubungkan dengan GitHub:** Menghubungkan repositori lokal dengan repositori GitHub menggunakan perintah git remote add origin <URL\_repo\_GitHub>.
* **Unggah Proyek:** Mengunggah proyek ke GitHub dengan perintah git push -u origin main, memastikan untuk mengganti main dengan nama branch utama yang digunakan.
* **Verifikasi dan Dokumentasi:** Memverifikasi bahwa proyek berhasil diunggah ke GitHub dengan melihat repositori di browser, dan menambahkan README file yang menjelaskan proyek, instalasi, dan cara menjalankannya.

Dengan mengunggah proyek ke GitHub, Anda tidak hanya memfasilitasi kolaborasi dengan tim, tetapi juga membangun reputasi sebagai pengembang yang terbuka dan proaktif dalam berbagi dan mengembangkan kode.