LAPORAN PEMROGRAMAN BERORIENTASI OBJEK

Dosen Pengampu :

I Gde Agung Sri Sidhimantra S.Kom., M.Kom.

Binti Kholifah, S.Kom., M.Tr.Kom.

Moch Deny Pratama, S.Tr.Kom., M.Kom.

Dimas Novian Aditia Syahputra, S.Tr.T., M.Tr.T.

Disusun Oleh :

Mohamat Fuat Hasan (23019397097)

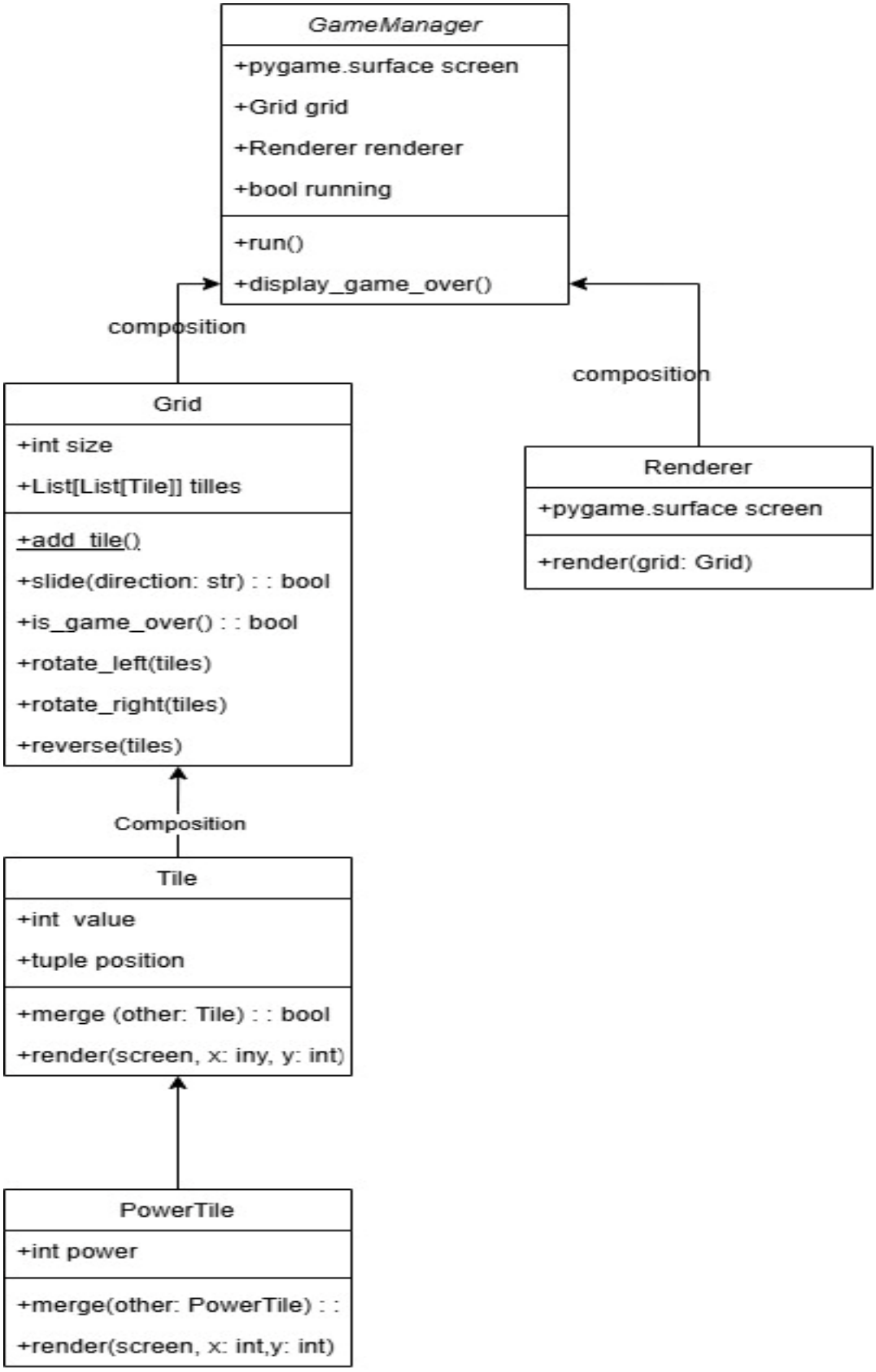
Dicky Dippos Sihite (203091397071)

Program Studi D4 Manajemen Informatika

Program Vokasi

Universitas Negeri Surabaya 2024

Diagram Class:



2048 Game

Diagram Kelas dan Struktur Kelas

1. Tile (Kelas Dasar)

Kelas `Tile` digunakan sebagai elemen dasar dari setiap blok dalam game.

Atribut:

- `value`: Nilai pada tile, seperti angka 2, 4, 8, dst.
- `position`: Posisi tile di dalam grid, direpresentasikan sebagai koordinat `(baris, kolom)`.

Metode Utama:

- `merge(other)`: Digunakan untuk menggabungkan dua tile dengan nilai yang sama. Setelah penggabungan, nilai tile akan menjadi dua kali lipat.
    - **Contoh:** Tile dengan nilai 2 digabungkan dengan tile lain yang bernilai 2, hasilnya tile bernilai 4.
- `render(screen, x, y)`: Menampilkan tile pada layar berdasarkan nilai dan posisinya.

2. PowerTile (Kelas Turunan dari Tile)

Kelas `PowerTile` memperluas fungsionalitas dari kelas `Tile` dengan menambahkan kemampuan khusus berupa "power".

Atribut Tambahan:

- `power`: Menyimpan nilai tambahan yang dapat meningkatkan kekuatan tile setiap kali tile digabungkan.

Modifikasi Metode:

- `merge(other)`: Selain menggandakan nilai tile seperti pada kelas induk, metode ini juga menambahkan nilai `power` setiap kali tile digabungkan.
- `render(screen, x, y)`: Sama seperti pada kelas `Tile`, tetapi ditambahkan tampilan nilai `power` sebagai elemen visual tambahan.

3. Grid

Kelas `Grid` bertanggung jawab untuk mengelola susunan tile di dalam grid permainan.

Atribut:

- `size`: Ukuran grid (4x4).

- `tiles`: Matriks dua dimensi untuk menyimpan semua tile di grid. Posisi yang belum terisi diwakili oleh `None`.

Fungsi Utama:

- `add_tile()`: Menambahkan tile baru di posisi kosong secara acak pada grid. Tile baru bisa berupa `Tile` biasa atau `PowerTile` dengan peluang tertentu.
- `slide(direction)`: Menggeser seluruh tile ke arah yang ditentukan pemain (atas, bawah, kiri, kanan). Jika ada tile dengan nilai yang sama berdekatan, mereka akan digabungkan.
- `is_game_over()`: Mengecek apakah permainan sudah berakhir. Kondisi akhir terjadi jika:
    - Tidak ada lagi ruang kosong di grid.
    - Tidak ada lagi tile yang dapat digabungkan.

4. Renderer

Kelas `Renderer` bertugas menggambar elemen permainan ke layar, termasuk grid dan tile.

Fungsi Utama:

- `render(grid)`: Menggambar seluruh grid permainan dan semua tile yang ada di dalamnya ke layar, termasuk menampilkan warna dan nilai masing-masing tile.

5. GameManager

Kelas `GameManager` adalah pengontrol utama yang mengatur seluruh alur permainan.

Atribut:

- `grid`: Objek dari kelas `Grid`, berfungsi untuk mengelola logika permainan.
- `renderer`: Objek dari kelas `Renderer`, digunakan untuk menggambar elemen permainan ke layar.
- `running`: Status permainan (berjalan atau tidak).

Fungsi Utama:

- `run()`:
    - Menangani input dari pemain (panah keyboard).
    - Memanggil metode pada grid untuk menggeser atau menggabungkan tile sesuai arah input.
    - Memperbarui tampilan layar menggunakan renderer.
    - Mengecek kondisi akhir permainan melalui `is_game_over()`.
- `display_game_over()`: Menampilkan pesan "Game Over" di layar setelah permainan selesai.

Hubungan Antar Kelas

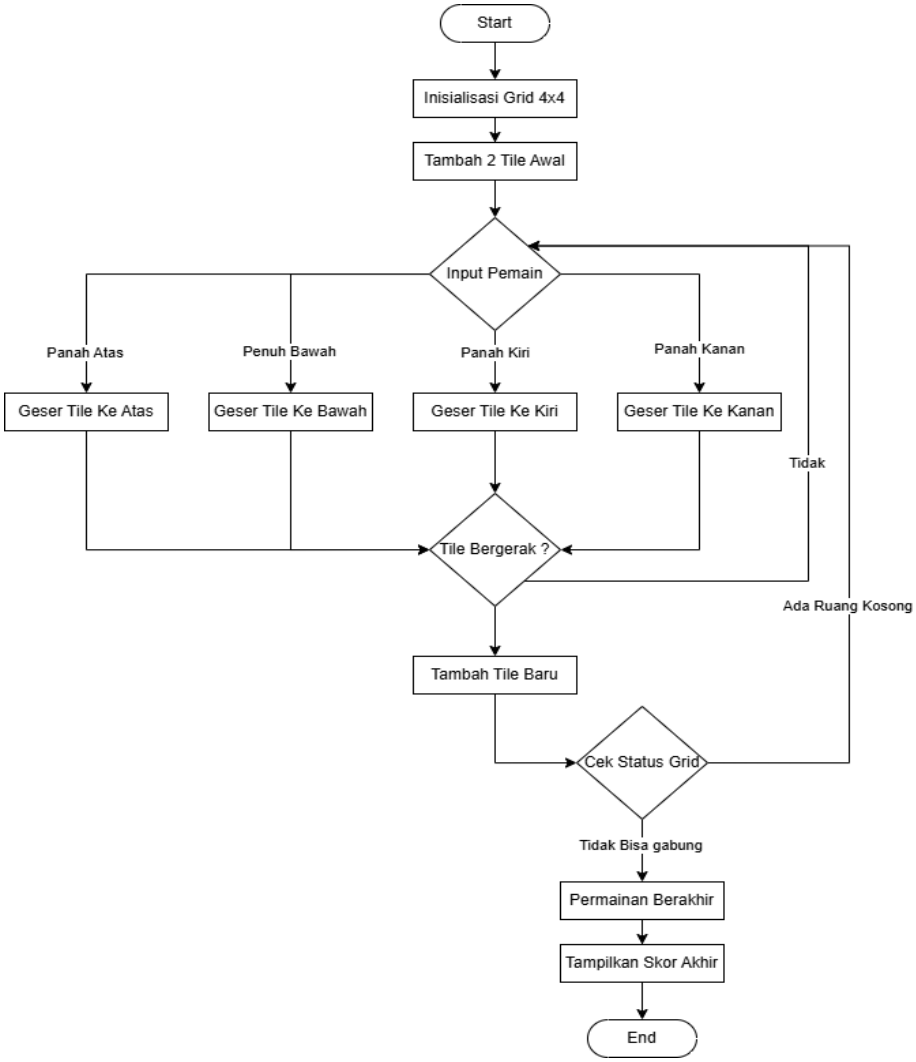
Pewarisan:

- Kelas `PowerTile` mewarisi kelas `Tile`, sehingga memiliki semua atribut dan metode kelas induk, dengan tambahan fungsi spesifik.

Komposisi:

- `GameManager` memiliki:
  - Objek dari kelas `Grid` untuk logika permainan.
  - Objek dari kelas `Renderer` untuk menggambar elemen ke layar.
- `Grid` berisi beberapa objek `Tile` dan `PowerTile`.

Flowchart dan Alur Permainan



Tahapan Permainan:

Inisialisasi:

1. Membuat grid ukuran 4x4 (16 posisi) menggunakan kelas `Grid`.
2. Menambahkan dua tile awal di posisi acak pada grid.

Input Pemain:

- Menunggu input pemain melalui tombol panah pada keyboard (atas, bawah, kiri, kanan).

Gerakan Tile:

1. Proses yang terjadi:
    - Tile digeser ke arah input.
    - Jika ada tile dengan nilai yang sama di jalur yang sama, mereka digabungkan.
2. Setelah setiap gerakan valid, tile baru akan ditambahkan secara acak.

Kondisi Akhir:

- Permainan dianggap selesai jika:
    - Tidak ada lagi posisi kosong pada grid.
    - Tidak ada lagi pasangan tile yang dapat digabungkan.
- Jika permainan selesai:
    - Pesan "Game Over" ditampilkan di layar.

Penggunaan OOP (Object-Oriented Programming)

Program ini memanfaatkan prinsip OOP secara luas. Berikut penjelasan terkait penerapan OOP:

1. Kelas dan Objek

- **Tile dan PowerTile**:
    - **Tile** adalah kelas dasar yang merepresentasikan sebuah ubin dalam permainan.
    - **PowerTile** adalah subclass dari Tile yang menambahkan atribut dan metode khusus (power). Ini adalah contoh **pewarisan (inheritance)** dan **polimorfisme (polymorphism)**.
- **Grid**:
    - Mengelola logika dan posisi ubin di grid. Grid terdiri dari matriks dua dimensi berisi objek Tile.
- **Renderer**:
    - Bertanggung jawab untuk menggambar semua elemen di layar. Setiap ubin diproses dan divisualisasikan menggunakan metode render.
- **GameManager**:
    - Mengatur alur permainan, seperti menerima input pemain, memeriksa kondisi permainan, dan memanggil renderer untuk menggambar ke layar.

2. Prinsip OOP yang Digunakan

- **Encapsulation (Enkapsulasi):**
    - Setiap kelas memiliki atribut dan metode yang terpisah sesuai fungsinya. Misalnya, atribut value dan position hanya tersedia dalam kelas Tile dan turunannya.
- **Inheritance (Pewarisan):**
    - PowerTile mewarisi semua atribut dan metode dari kelas Tile, sehingga dapat memperluas fungsionalitas tanpa harus menduplikasi kode.
- **Polymorphism (Polimorfisme):**
    - Metode merge dan render pada kelas PowerTile **meng-override** metode yang sama dari kelas Tile untuk memberikan perilaku unik.
- **Abstraction (Abstraksi):**
    - GameManager mengabstraksi alur permainan sehingga logika game tetap terorganisir dan mudah dimengerti.

Fitur Aplikasi

1. **Pengelolaan Grid**:
    - Grid 4x4 secara dinamis menambah ubin baru setelah setiap gerakan pemain.
    - Sistem otomatis menggabungkan ubin dengan nilai yang sama.
2. **Tile Biasa dan PowerTile**:
    - **Tile** adalah ubin standar dengan nilai tertentu.
    - **PowerTile** menambahkan fitur tambahan berupa "power", memberikan variasi dalam gameplay.
3. **Gerakan Dinamis**:
    - Ubin dapat digeser ke atas, bawah, kiri, atau kanan. Setiap pergeseran menggabungkan ubin yang memiliki nilai sama.
4. **Pemeriksaan Akhir Permainan**:
    - Game akan menampilkan layar "Game Over" jika tidak ada lagi gerakan yang memungkinkan.
5. **Visualisasi**:
    - Tiap ubin memiliki warna dan nilai berbeda berdasarkan nilainya. PowerTile memiliki informasi tambahan tentang power.

Cara Menggunakan Aplikasi

1. **Menjalankan Program**:
    - Jalankan file menggunakan Python dengan pustaka **pygame** terinstal.
2. **Kontrol Permainan**:
    - Gunakan tombol panah **(↑, ↓, ←, →)** untuk menggeser ubin ke arah tertentu.
3. **Mekanisme Permainan**:
    - Ubin akan bergabung jika memiliki nilai yang sama.
    - Jika grid penuh dan tidak ada gerakan yang memungkinkan, permainan berakhir.
4. **Game Over**:
    - Saat permainan selesai, layar akan menampilkan teks "Game Over" selama 3 detik sebelum keluar dari aplikasi.
