

Bab 7

TiledLayer

7.1 Tujuan

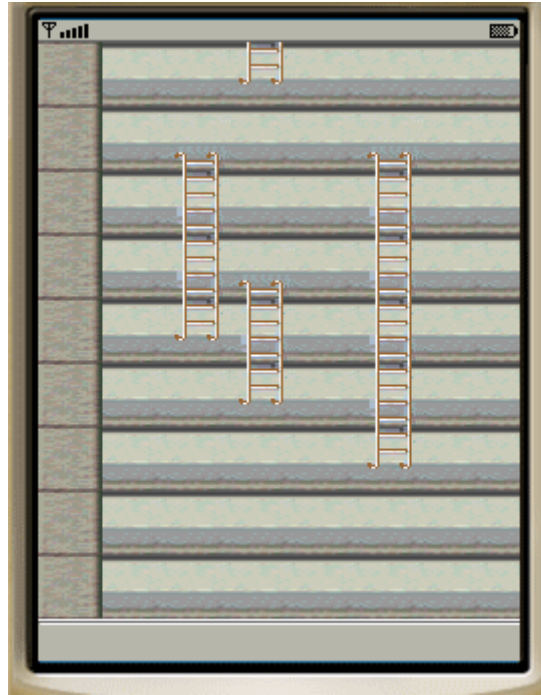
Setelah mempelajari bab ini, Pelajar diharapkan mampu untuk :

- Memahami dan menggunakan class TiledLayer
- Menggambar dan menyusun TiledLayer
- Menyusun graphics dari TiledLayer dan di masukkan dalam LayerManager
- Memanipulasi graphics dengan menggunakan method yang ada pada class TiledLayer

7.2 Pengenalan

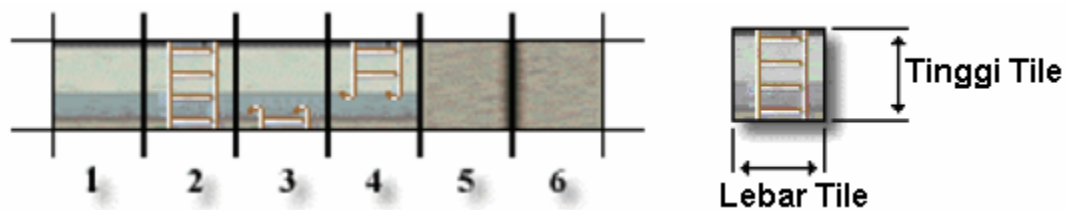
Games dengan background yang besar/area virtual pada TiledLayer adalah suatu yang sempurna untuk berhubungan dengan kelas ini. TiledLayer akan berfungsi untuk mendefinisikan semua area unik pada background dan menggunakannya kembali sebanyak yang diperlukan untuk menciptakan sebuah gambar yang lengkap.

7.3 Penggunaan TiledLayer



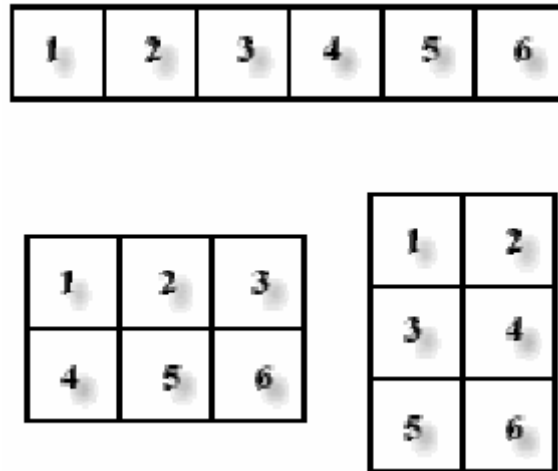
Gambar 1: Contoh TiledLayer

Pada gambar 1, jika Anda mempelajari gambar tersebut Anda akan menemukan beberapa area yang mirip dan kenyataannya Anda dapat memecah gambar tersebut ke dalam 6 area terpisah.



Gambar 2: Contoh sistem potongan pada Tiled Layer

Anda akan menemukan semua tile memiliki dimensi yang sama yaitu 32 x 32 pixel yang diperlukan oleh kelas `TiledLayer`. Setiap tile diberikan nilai indeks yang dimulai dari 1. Angka yang ada dimulai dari kiri ke kanan dan atas ke bawah. Dengan kata lain angka ditetapkan dengan berbasis baris per baris. Anda dapat meletakkan tile pada berbagai konfigurasi dan dengan asumsi bahwa Anda dapat memastikan bahwa nilai indeks yang ditetapkan tidak akan berubah contoh pada gambar 3 dibawah ini.



Gambar 2: Konfigurasi penempatan tiledLayer

7.3.1 Penetapan indeks TiledLayer

Contoh diatas tile dianggap tile yang statis. Jika indeks berisi nilai nol ini mengindikasikan bahwa cell tersebut adalah kosong, yang berarti setiap cell yang bernilai nol tidak akan menggambar apapun pada area yang ditempati.

7.3.2 Konstruktor TiledLayer

Untuk membuat sebuah `TiledLayer` maka harus meng-instantiate konstruktor :

```
new TiledLayer(int columns, int rows, Image image, int tileWidth, int tileHeight);
```

2 nilai pertama menunjuk pada kolom dan baris yang dapat menyusun gambar sampai selesai. Parameter ketiga adalah gambar dari lembaran tile yang digunakan pada map yang berlawanan dengan nilai indeks. Sedangkan 2 parameter terakhir adalah lebar dan tinggi saat ini untuk tile tertentu.

7.3.3 Manipulasi TiledLayer

Untuk memetakan sebuah tile gambar pada sebuah cell tertentu Anda perlu menggunakan method `setCell(int col, int row, int tileIndex)`. Kolom dan baris adalah lokasi dimana Anda ingin me-render tile gambar dan parameter terakhir pada pemetaan tile akan di-render pada lokasi cell tersebut.

Untuk menimpa seluruh tile gunakan method `setStaticTileSet(Image image, int tileWidth, tileHeight)`. Jika tile set baru sama atau berisi lebih banyak dari original tile set, tile yang dianimasikan dan isi cell akan berjumlah sama pula. Walaupun demikian, jika isi tile baru lebih sedikit daripada tile set sebelumnya, maka pemetaan akan di-reset, semua indeks akan berisi nilai nol dan semua tile animasi akan dihapus.

Untuk mengisi daerah cell tertentu dengan tile tertentu Anda dapat menggunakan method `fillCells(int col, int row, int numCols, int numRows, int tileIndex)`. Nilai pertama mengindikasikan kolom dan baris untuk cell kiri atas pada daerah yang telah ditentukan. Yang kedua untuk mendefinisikan jumlah kolom dan baris. Parameter yang terakhir adalah indeks yang ingin Anda petakan/letakkan.

7.3.4 Tampilan TiledLayer

Seperti semua objek game lainnya, Anda perlu memanggil method `paint` secara langsung atau menggunakan `LayerManager`.

Mendapatkan Kembali Pengaturan `TiledLayer`

Ada beberapa method yang tersedia untuk mendapatkan informasi pada penggunaan `TiledLayer`:

`getCell(int col, int row)`

Mengambil semua isi pada cell.

`getCellHeight()`

Mengambil tinggi dari 1 cell dalam bentuk piksel.

`getCellWidth()`

Mengambil width dari 1 cell dalam bentuk piksel.

`getColumns()`

Mengambil jumlah kolom di dalam tiledlayer.

`getRows()`

Mengambil jumlah baris di dalam tiledlayer.

7.3.4 Menggambar TiledLayer

Ada satu fitur lagi dari TiledLayer yaitu animated tiles. Animated tiles diindikasikan dengan nilai indeks negatif. Setiap animated cell secara dinamis diasosiasikan menjadi sebuah tile yang statik. Ini memungkinkan kita agar lebih mudah menganimasikan sebuah grup cell dengan mengubah cell statik yang diasosiasikan. Ini bagus sekali untuk animasi background ramai, awan bergerak dan/atau riak air. Terlepas dari sebelumnya method yang disebutkan ada tiga method tambahan khusus untuk animated cell.

- `createAnimatedTile(int staticTileIndex)` – membuat sebuah animated cell baru pada indeks yang ditentukan dan mengembalikan urutan index negatif berikutnya untuk animated tile. Secara default berisi sebuah tile statik (angka positif) atau nilai nol.
- `getAnimatedTile(int animatedTileIndex)` – mendapatkan kembali pemetaan tile pada indeks animated tile.
- `setAnimatedtile(int animatedTileIndex, int staticTileIndex)` – menghubungkan sebuah animated tile pada sebuah statik tile.

Hasilnya pada gambar 1 diatas.

TiledLayerCanvas Source Code:

.....

```
public class TiledLayerCanvas extends GameCanvas implements Runnable
{
    // Penggunaan Sprites
    private Sprite playerSprite;
```

```

// Layer Manager
private LayerManager layerManager;

// TiledLayer
private TiledLayer tiledBackground;

// Konstruktor dan inisialisasi
public ExampleGameCanvas() throws Exception {
    .....
    tiledBackground = initBackground();
    layerManager = new LayerManager();
    layerManager.append(tiledBackground);
}

// Method yang membuat Tiled Layer
private TiledLayer initBackground() throws Exception {
    Image tileImages = Image.createImage("/tiles.png");
    TiledLayer tiledLayer =
        new TiledLayer(10,10,tileImages,32,32);
    int[] map = {
        5, 1, 1, 4, 1, 1, 1, 1, 1, 6,
        5, 1, 3, 1, 1, 3, 1, 1, 1, 6,
        5, 1, 2, 1, 1, 2, 1, 1, 1, 6,
        5, 1, 2, 3, 1, 2, 1, 1, 1, 6,
        5, 1, 4, 2, 1, 2, 1, 1, 1, 6,
        5, 1, 1, 4, 1, 2, 1, 1, 1, 6,
        5, 1, 1, 1, 1, 4, 1, 1, 1, 6,
        5, 1, 1, 1, 1, 1, 1, 1, 1, 6,
        5, 1, 1, 1, 1, 1, 1, 1, 1, 6,
        5, 1, 1, 1, 1, 1, 1, 1, 1, 6
    };

    for (int i=0; i < map.length; i++) {
        int column = i % 10;
        int row = (i - column) / 10;
        tiledLayer.setCell(column,row,map[i]);
    }
    return tiledLayer;
}
}

```

7.3.5 Menggambar Animasi TiledLayer

Animasi TiledLayer adalah menggantikan potongan layer dari index a ke index b misalnya pada layer index 2 di rubah menjadi index 3 animasi ini akan bergerak terus per delaynya

AnimationTiledLayerCanvas Source Code:

```
public class AnimationTiledLayerCanvas extends GameCanvas implements
Runnable {
    .....

    // Method untuk menampilkan Graphic
    private void drawScreen(Graphics g) {
        .....
        // Menentukan tile mana yang akan ditampilkan
        if (switchTile) {
            tiledBackground.setAnimatedTile(animatedIdx,3);
        } else {
            tiledBackground.setAnimatedTile(animatedIdx,4);
        }

        // men-set tile menjadi nilai boolean
        switchTile = !switchTile;
        layerManager.paint(g,0,0);
        flushGraphics();
    }

    private TiledLayer initBackground() throws Exception {
        Image tileImages = Image.createImage("/tiles.png");
        .....

        for (int i=0; i < map.length; i++) {
            .....
        }

        // Menciptakan animate tile dan mendapatkan indeks animated tile
        animatedIdx = tiledLayer.createAnimatedTile(5);

        // men-set Cell dengan indeks animated tile
        tiledLayer.setCell(1,1,animatedIdx);
        return tiledLayer;
    }
}
```

**Animasi antara potongan atas pada tangga dengan
potongan bawah pada tangga**



Gambar 3: Animasi TiledLayer