



UAS

PEMOGRAMAN BERORIENTASI OBJEK

Penyusun :

MUHAMMAD WAHYU HIDAYATULLAH
(2211102441148)

MUHAMMAD FAUZAN ANANDA SYALTRI
(2211102441156)

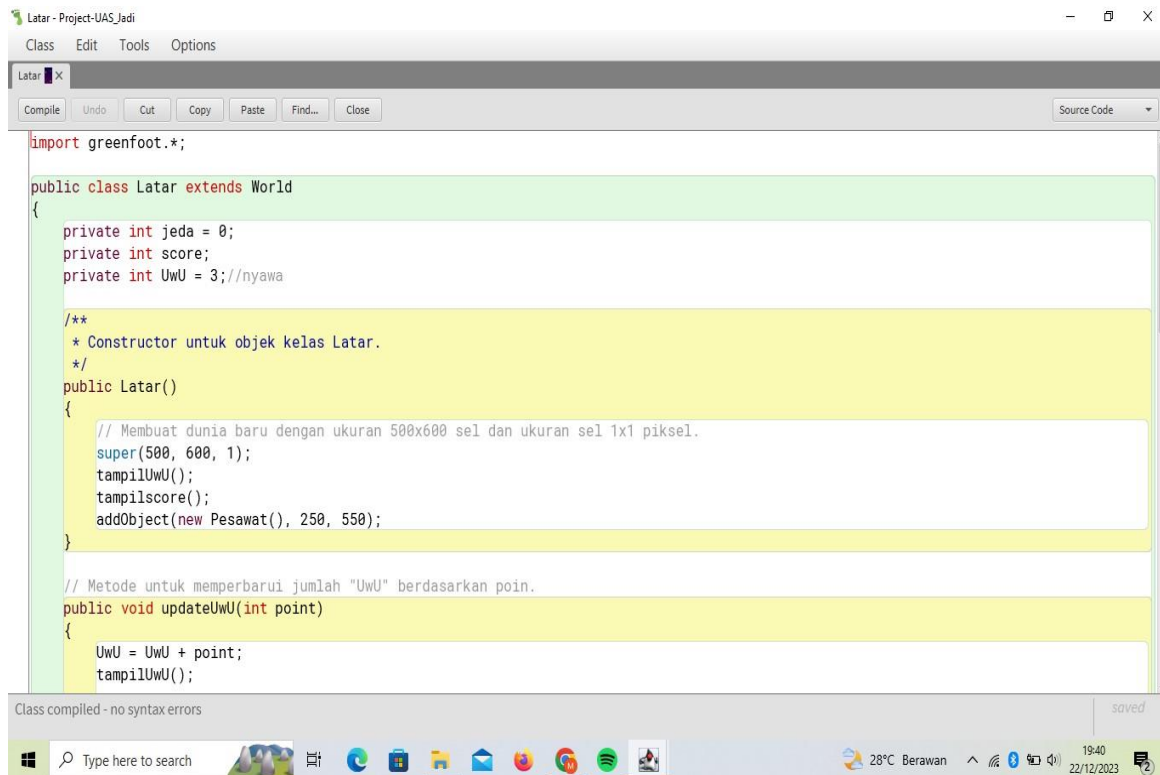
MUH.RYAMIZAR SYAFAR
(2211102441146)

Teknik Informatika
Fakultas Sains & Teknologi
Universitas Muhammadiyah Kalimantan Timur
Samarinda, 22 Desember 2023

- **Pembuka**

Kami telah mengembangkan sebuah game Pesawat tempur sederhana menggunakan Greenfoot, di mana pemain dapat mengendalikan pesawat untuk menghancurkan musuh dan mencapai skor tertentu. Game ini melibatkan elemen UwU sebagai nyawa, dengan tujuan untuk mencapai kemenangan sebelum kehabisan nyawa. Berikut adalah laporan singkat mengenai konsep, fitur, dan pengalaman pengguna dari game yang telah saya buat.

1. LATAR



```
import greenfoot.*;

public class Latar extends World
{
    private int jeda = 0;
    private int score;
    private int UwU = 3; //nyawa

    /**
     * Constructor untuk objek kelas Latar.
     */
    public Latar()
    {
        // Membuat dunia baru dengan ukuran 500x600 sel dan ukuran sel 1x1 piksel.
        super(500, 600, 1);
        tampilUwU();
        tampilScore();
        addObject(new Pesawat(), 250, 550);
    }

    // Metode untuk memperbarui jumlah "UwU" berdasarkan poin.
    public void updateUwU(int point)
    {
        UwU = UwU + point;
        tampilUwU();
    }
}
```

Class compiled - no syntax errors

Windows taskbar: 28°C Berawan, 19:40, 22/12/2023

```
UwU = UwU + point;
tampilUwU();

// Jika "UwU" mencapai 0, hentikan permainan dan tambahkan layar kalah.
if(UwU == 0)
{
    Greenfoot.stop();
    addObject(new kalah(), 250, 300);
}

// Metode untuk menampilkan jumlah "UwU" di layar.
public void tampilUwU()
{
    showText("UwU :" + UwU, 400, 25);
}

// Metode untuk menambah skor dan menampilkan perubahan di layar.
public void addScore(int points)
{
    score = score + points;
    tampilscore();

    // Tambahkan kondisi untuk menang: jika skor mencapai 20, hentikan permainan dan tambahkan layar menang.
    if (score >= 20)
    {
        Greenfoot.stop();
        addObject(new Menang(), 250, 300);
    }
}
```

Class compiled - no syntax errors

```
// Tambahkan kondisi untuk menang: jika skor mencapai 20, hentikan permainan dan tambahkan layar menang.
if (score >= 20)
{
    Greenfoot.stop();
    addObject(new Menang(), 250, 300);
}

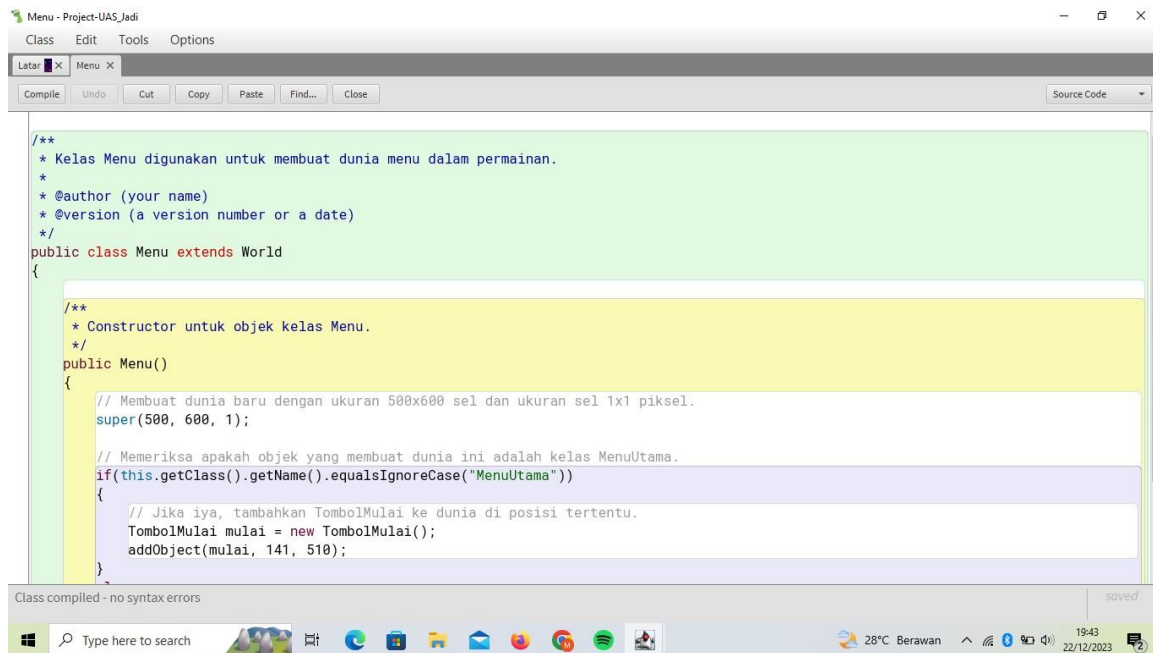
// Metode untuk menampilkan skor di layar.
private void tampilscore()
{
    showText("Score :" + score, 100, 25);
}

// Metode act dipanggil pada setiap frame dan digunakan untuk mengatur logika permainan.
public void act()
{
    // Jika angka acak kurang dari 1 (1% kemungkinan), tambahkan objek Musuh ke dunia.
    if (Greenfoot.getRandomNumber(100) < 1)
    {
        addObject(new Musuh(), Greenfoot.getRandomNumber(599), 500);
    }
}
```

Class compiled - no syntax errors

- Kelas Latar dalam permainan Greenfoot mengelola logika permainan, termasuk nyawa pemain (UwU), skor, dan penambahan musuh ke dunia. Jika nyawa habis, layar kalah ditampilkan. Jika skor mencapai 20, layar menang ditampilkan. Musuh ditambahkan ke dunia secara acak dengan probabilitas 1%.

2. MENU

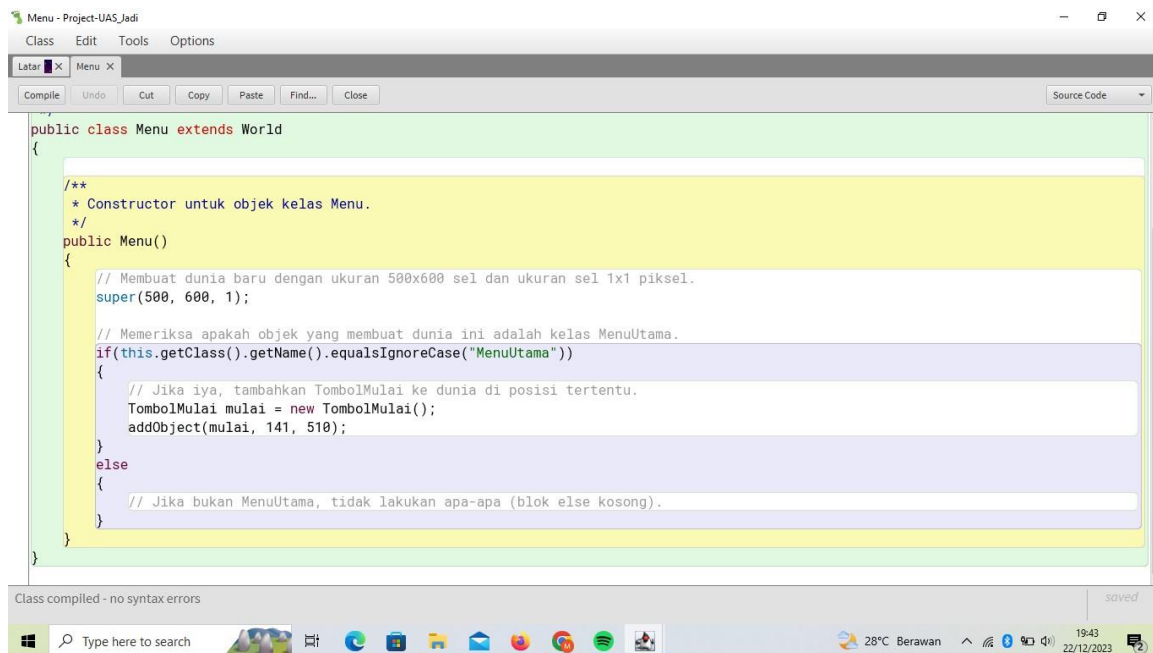


The screenshot shows an IDE window titled "Menu - Project-UAS_Jadi". The menu bar includes "Class", "Edit", "Tools", and "Options". Below the menu bar is a toolbar with buttons for "Compile", "Undo", "Cut", "Copy", "Paste", "Find...", and "Close". The main editor area displays the following Java code:

```
/**
 * Kelas Menu digunakan untuk membuat dunia menu dalam permainan.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class Menu extends World
{
    /**
     * Constructor untuk objek kelas Menu.
     */
    public Menu()
    {
        // Membuat dunia baru dengan ukuran 500x600 sel dan ukuran sel 1x1 piksel.
        super(500, 600, 1);

        // Memeriksa apakah objek yang membuat dunia ini adalah kelas MenuUtama.
        if(this.getClass().getName().equalsIgnoreCase("MenuUtama"))
        {
            // Jika iya, tambahkan TombolMulai ke dunia di posisi tertentu.
            TombolMulai mulai = new TombolMulai();
            addObject(mulai, 141, 510);
        }
    }
}
```

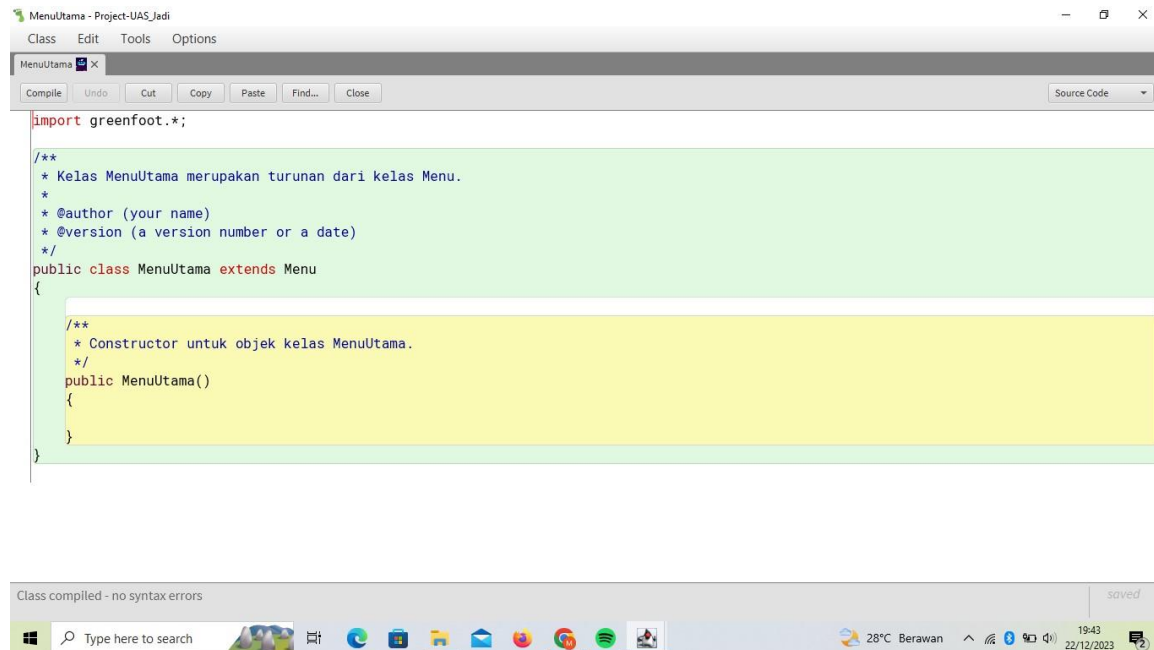
At the bottom of the IDE window, a status bar indicates "Class compiled - no syntax errors" and "saved". The Windows taskbar at the bottom shows the search bar, task view, and several application icons, along with system tray information: "28°C Berawan", "19:43", and "22/12/2023".



This screenshot is identical to the one above, showing the same Java code for the Menu class in the IDE. It includes the same menu bar, toolbar, code editor content, and status bar information.

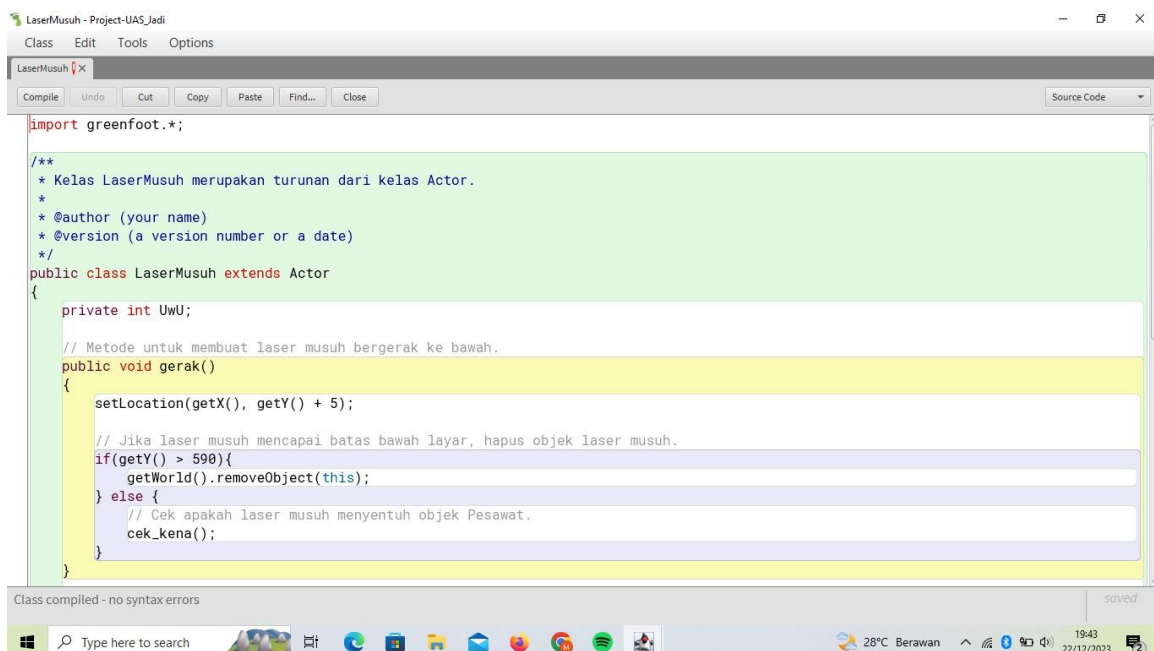
- Kelas Menu bertindak sebagai dunia menu dalam permainan. Saat dunia ini dibuat, itu memiliki ukuran 500x600 sel dengan ukuran sel 1x1 piksel. Jika objek yang membuat dunia ini adalah kelas MenuUtama, maka tambahkan tombol mulai (TombolMulai) ke dunia pada posisi tertentu (141, 510). Jika bukan MenuUtama, tidak ada tindakan yang diambil (blok else kosong).

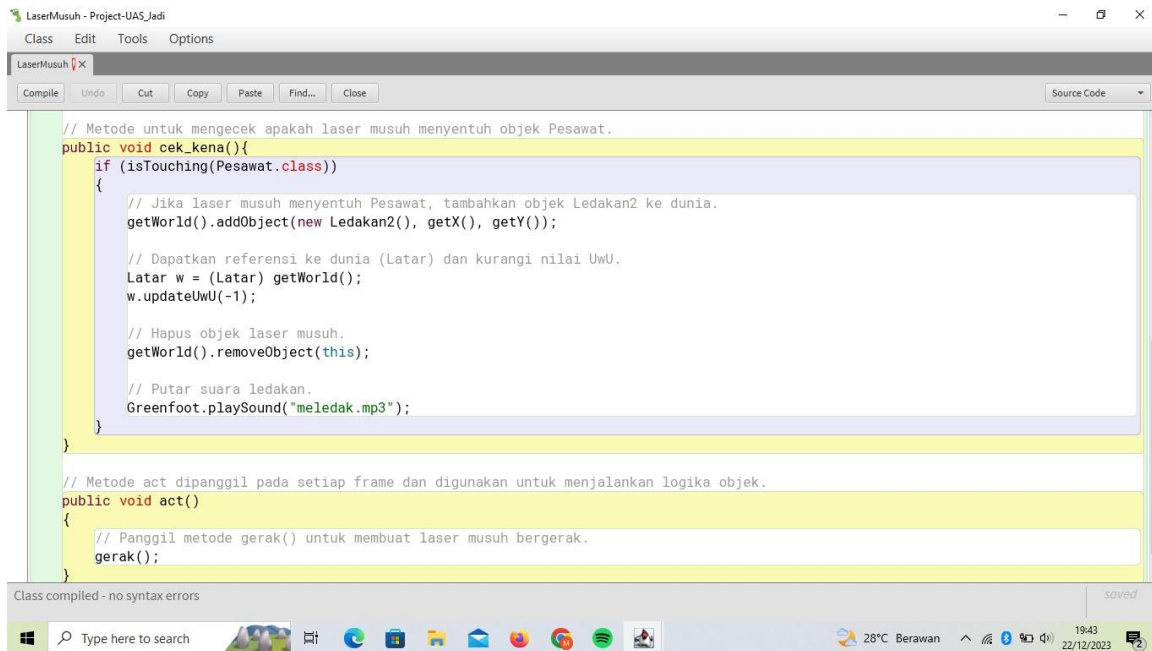
3. MENU UTAMA



- Kelas `MenuUtama` merupakan turunan dari kelas `Menu`. Saat dunia ini dibuat, tidak ada tindakan tambahan yang diambil di dalam konstruktor. Jadi, kelas ini hanya mewarisi konstruktor dari kelas induknya (`Menu`) tanpa menambahkan perilaku tambahan.

4. LASER MUSUH





```
// LaserMusuh - Project-UAS_Jadi
// Class Edit Tools Options
// LaserMusuh
// Compile Undo Cut Copy Paste Find... Close Source Code

// Metode untuk mengecek apakah laser musuh menyentuh objek Pesawat.
public void cek_kena(){
    if (isTouching(Pesawat.class))
    {
        // Jika laser musuh menyentuh Pesawat, tambahkan objek Ledakan2 ke dunia.
        getWorld().addObject(new Ledakan2(), getX(), getY());

        // Dapatkan referensi ke dunia (Latar) dan kurangi nilai UwU.
        Latar w = (Latar) getWorld();
        w.updateUwU(-1);

        // Hapus objek laser musuh.
        getWorld().removeObject(this);

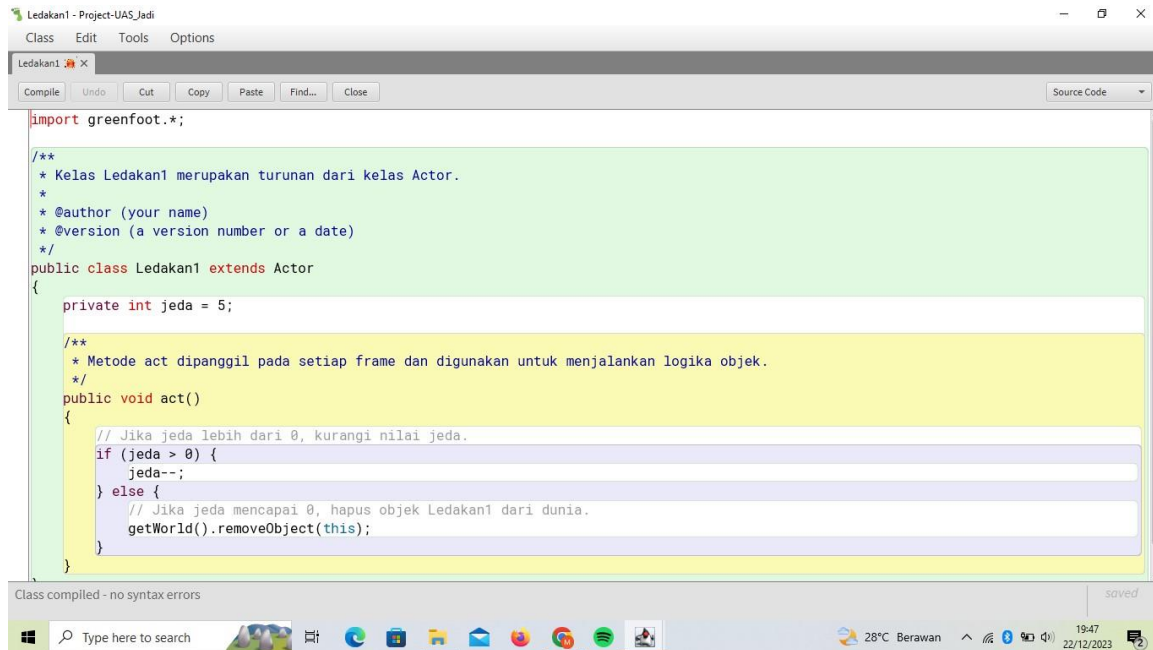
        // Putar suara ledakan.
        Greenfoot.playSound("meledak.mp3");
    }
}

// Metode act dipanggil pada setiap frame dan digunakan untuk menjalankan logika objek.
public void act()
{
    // Panggil metode gerak() untuk membuat laser musuh bergerak.
    gerak();
}

Class compiled - no syntax errors
saved
19:43 22/12/2023
```

- Kelas LaserMusuh merupakan turunan dari kelas Actor. Ketika objek laser musuh dijalankan (act dipanggil pada setiap frame), objek tersebut akan bergerak ke bawah dengan memanggil metode gerak(). Jika laser musuh mencapai batas bawah layar, objek tersebut akan dihapus dari dunia.
- Selain itu, terdapat metode cek_kena() yang digunakan untuk mengecek apakah laser musuh menyentuh objek Pesawat. Jika iya, maka akan ditambahkan objek Ledakan2 ke dunia, nilai UwU (nyawa) akan dikurangi satu, dan objek laser musuh akan dihapus. Suara ledakan juga akan diputar.
- Metode act() dipanggil pada setiap frame dan menjalankan metode gerak() untuk membuat laser musuh bergerak.

5. LEDAKAN 1



```
import greenfoot.*;

/**
 * Kelas Ledakan1 merupakan turunan dari kelas Actor.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class Ledakan1 extends Actor
{
    private int jeda = 5;

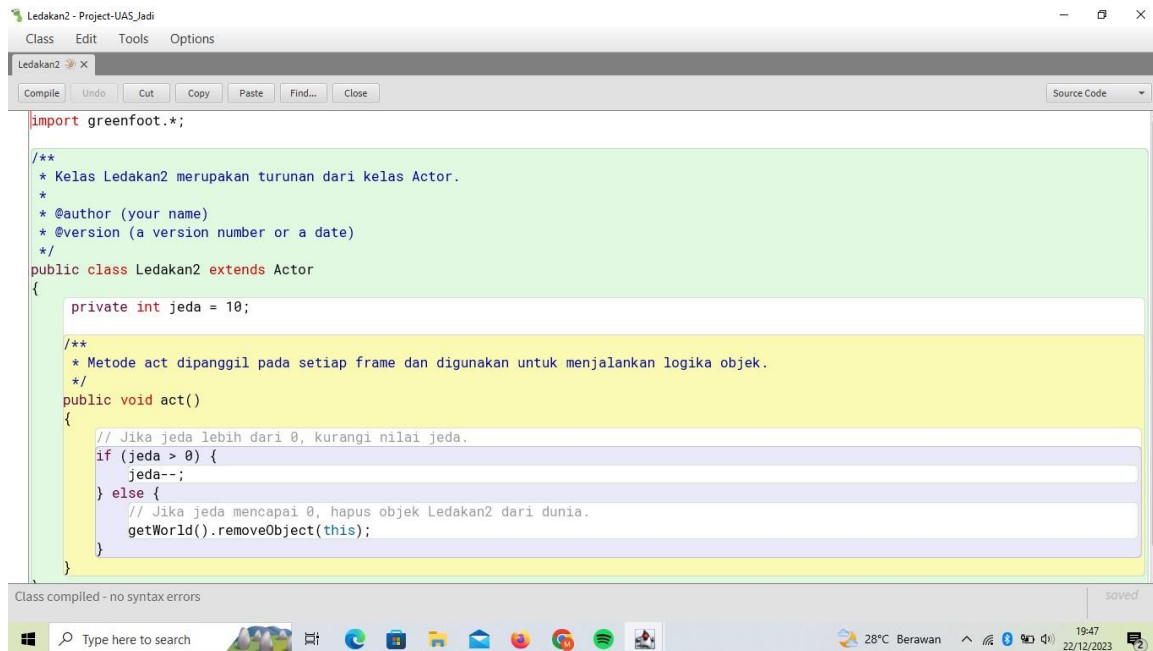
    /**
     * Metode act dipanggil pada setiap frame dan digunakan untuk menjalankan logika objek.
     */
    public void act()
    {
        // Jika jeda lebih dari 0, kurangi nilai jeda.
        if (jeda > 0) {
            jeda--;
        } else {
            // Jika jeda mencapai 0, hapus objek Ledakan1 dari dunia.
            getWorld().removeObject(this);
        }
    }
}
```

Class compiled - no syntax errors

28°C Berawan 19:47 22/12/2023

- Kelas Ledakan1 merupakan turunan dari kelas Actor. Ketika objek ini dijalankan (act dipanggil pada setiap frame), terdapat variabel jeda yang digunakan untuk mengatur waktu tampilan ledakan. Jika nilai jeda lebih dari 0, nilai jeda akan dikurangi setiap frame. Setelah jeda mencapai 0, objek Ledakan1 akan dihapus dari dunia.

6. LEDAKAN 2



```
import greenfoot.*;

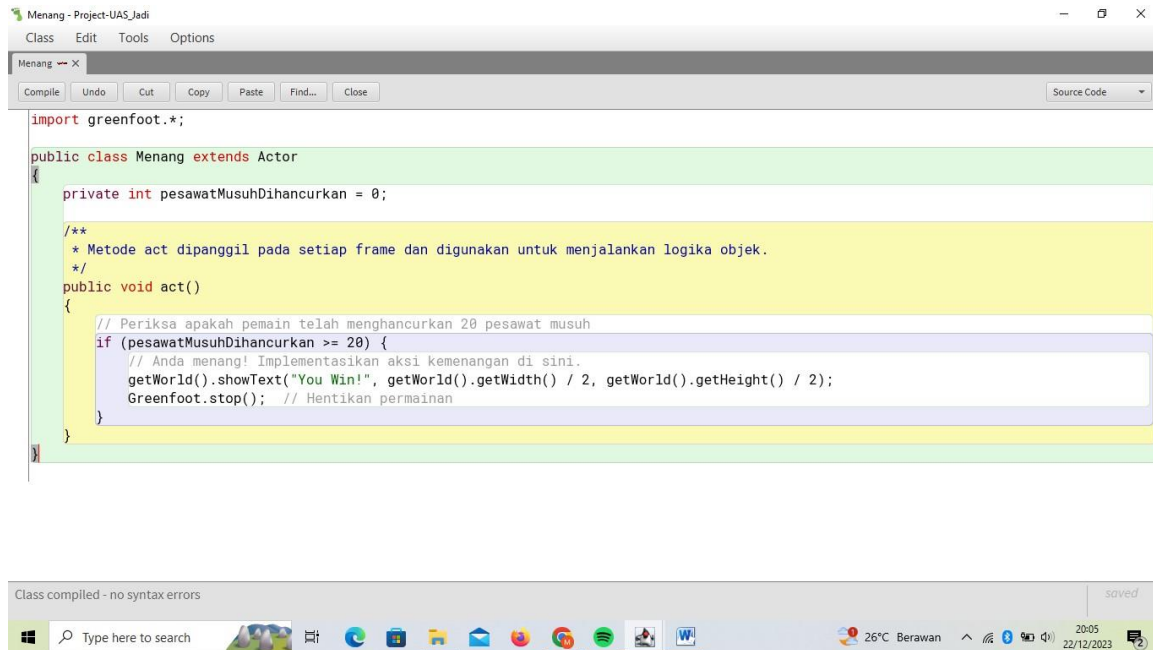
/**
 * Kelas Ledakan2 merupakan turunan dari kelas Actor.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class Ledakan2 extends Actor
{
    private int jeda = 10;

    /**
     * Metode act dipanggil pada setiap frame dan digunakan untuk menjalankan logika objek.
     */
    public void act()
    {
        // Jika jeda lebih dari 0, kurangi nilai jeda.
        if (jeda > 0) {
            jeda--;
        } else {
            // Jika jeda mencapai 0, hapus objek Ledakan2 dari dunia.
            getWorld().removeObject(this);
        }
    }
}
```

Class compiled - no syntax errors

- Kelas Ledakan2 juga merupakan turunan dari kelas Actor. Seperti Ledakan1, objek ini memiliki variabel jeda yang digunakan untuk mengatur waktu tampilan ledakan. Pada setiap frame, nilai jeda dikurangi. Ketika nilai jeda mencapai 0, objek Ledakan2 akan dihapus dari dunia.

7. MENANG



The screenshot shows a Java IDE window titled "Menang - Project-UAS_Jadi". The code editor displays the following Java code for the 'Menang' class, which extends 'Actor'.

```
import greenfoot.*;

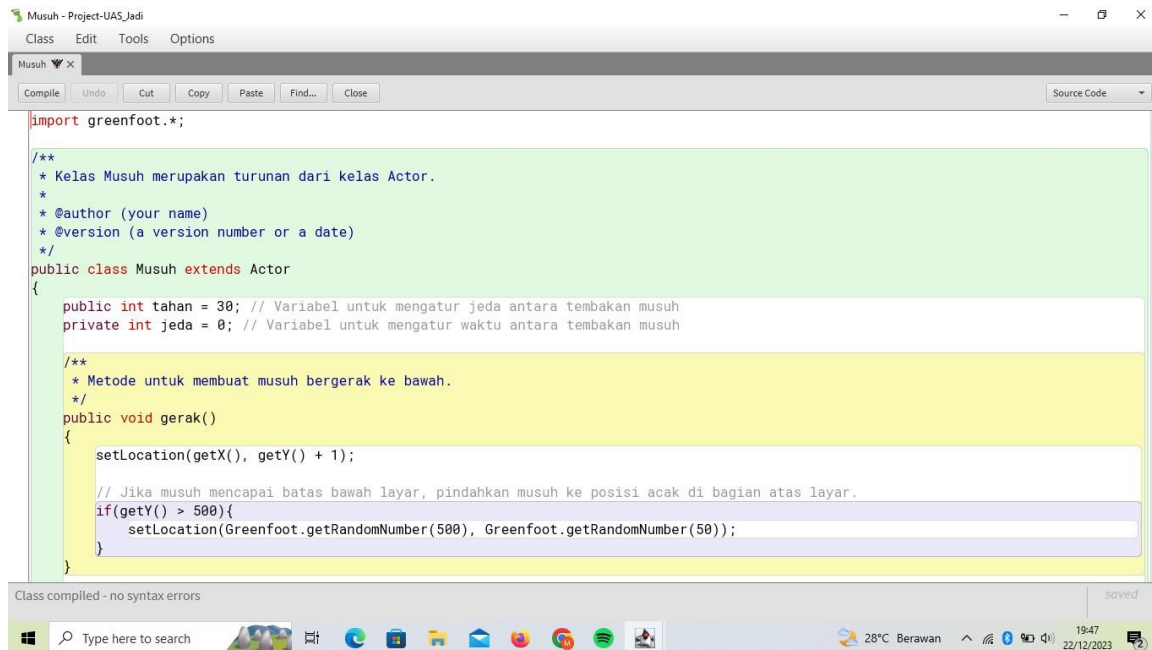
public class Menang extends Actor
{
    private int pesawatMusuhDihancurkan = 0;

    /**
     * Metode act dipanggil pada setiap frame dan digunakan untuk menjalankan logika objek.
     */
    public void act()
    {
        // Periksa apakah pemain telah menghancurkan 20 pesawat musuh
        if (pesawatMusuhDihancurkan >= 20) {
            // Anda menang! Implementasikan aksi kemenangan di sini.
            getWorld().showText("You Win!", getWorld().getWidth() / 2, getWorld().getHeight() / 2);
            Greenfoot.stop(); // Hentikan permainan
        }
    }
}
```

Below the code editor, a status bar indicates "Class compiled - no syntax errors" and "saved". The Windows taskbar at the bottom shows the system clock as 20:05 on 22/12/2023, with a temperature of 26°C in Berawan.

- Kelas Menang memiliki variabel pesawatMusuhDihancurkan untuk melacak jumlah pesawat musuh yang dihancurkan. Pada setiap frame, jika jumlah tersebut mencapai 20, pesan "You Win!" ditampilkan di tengah layar, dan permainan dihentikan.

8. MUSUH



```
import greenfoot.*;

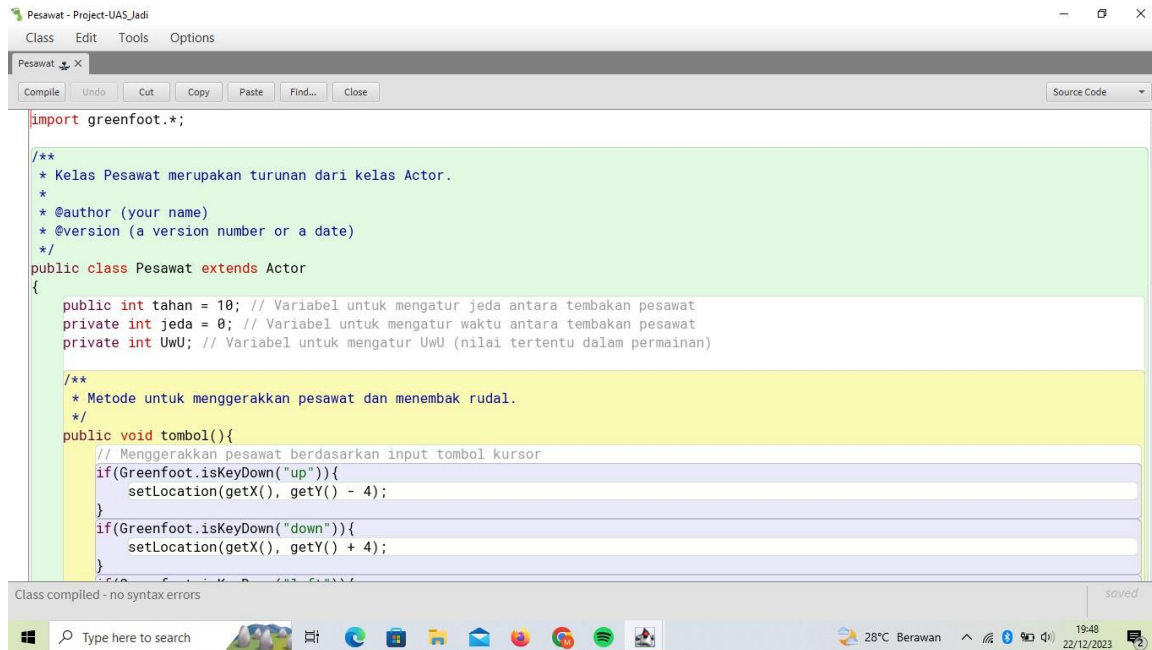
/**
 * Kelas Musuh merupakan turunan dari kelas Actor.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class Musuh extends Actor
{
    public int tahan = 30; // Variabel untuk mengatur jeda antara tembakan musuh
    private int jeda = 0; // Variabel untuk mengatur waktu antara tembakan musuh

    /**
     * Metode untuk membuat musuh bergerak ke bawah.
     */
    public void gerak()
    {
        setLocation(getX(), getY() + 1);

        // Jika musuh mencapai batas bawah layar, pindahkan musuh ke posisi acak di bagian atas layar.
        if(getY() > 500){
            setLocation(Greenfoot.getRandomNumber(500), Greenfoot.getRandomNumber(500));
        }
    }
}
```

- Kelas Musuh memiliki variabel tahan untuk mengatur jeda antara tembakan musuh dan variabel jeda untuk mengatur waktu antara tembakan musuh. Metode gerak() membuat musuh bergerak ke bawah, dan metode act() dipanggil pada setiap frame untuk menjalankan logika objek Musuh.
- Pada setiap frame, Musuh akan bergerak ke bawah. Ketika nilai tahan mencapai 0, nilai diatur kembali, dan jika nilai jeda lebih dari 0, dikurangi. Ketika nilai jeda mencapai 1, laser musuh ditambahkan ke dunia. Setelah itu, nilai jeda diatur kembali ke 120 untuk menunggu waktu sebelum tembakan musuh selanjutnya.

9. PESAWAT



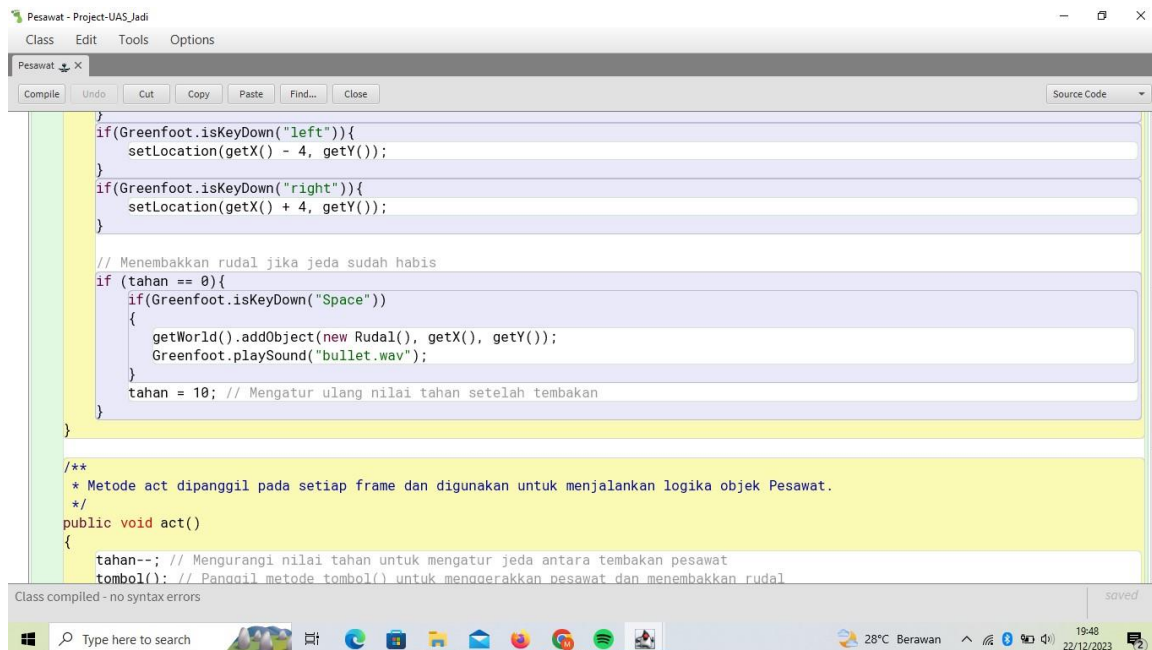
The screenshot shows a Java IDE window titled "Pesawat - Project-UAS_Jadi". The menu bar includes "Class", "Edit", "Tools", and "Options". The toolbar has buttons for "Compile", "Undo", "Cut", "Copy", "Paste", "Find...", and "Close". The "Source Code" dropdown is visible. The code in the editor is as follows:

```
import greenfoot.*;

/**
 * Kelas Pesawat merupakan turunan dari kelas Actor.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class Pesawat extends Actor
{
    public int tahan = 10; // Variabel untuk mengatur jeda antara tembakan pesawat
    private int jeda = 0; // Variabel untuk mengatur waktu antara tembakan pesawat
    private int UwU; // Variabel untuk mengatur UwU (nilai tertentu dalam permainan)

    /**
     * Metode untuk menggerakkan pesawat dan menembak rudal.
     */
    public void tombol(){
        // Menggerakkan pesawat berdasarkan input tombol kursor
        if(Greenfoot.isKeyDown("up")){
            setLocation(getX(), getY() - 4);
        }
        if(Greenfoot.isKeyDown("down")){
            setLocation(getX(), getY() + 4);
        }
    }
}
```

At the bottom of the IDE, a status bar indicates "Class compiled - no syntax errors" and "saved". The Windows taskbar at the very bottom shows the search bar, several application icons, and system information: "28°C Berawan", "19:48", and "22/12/2023".



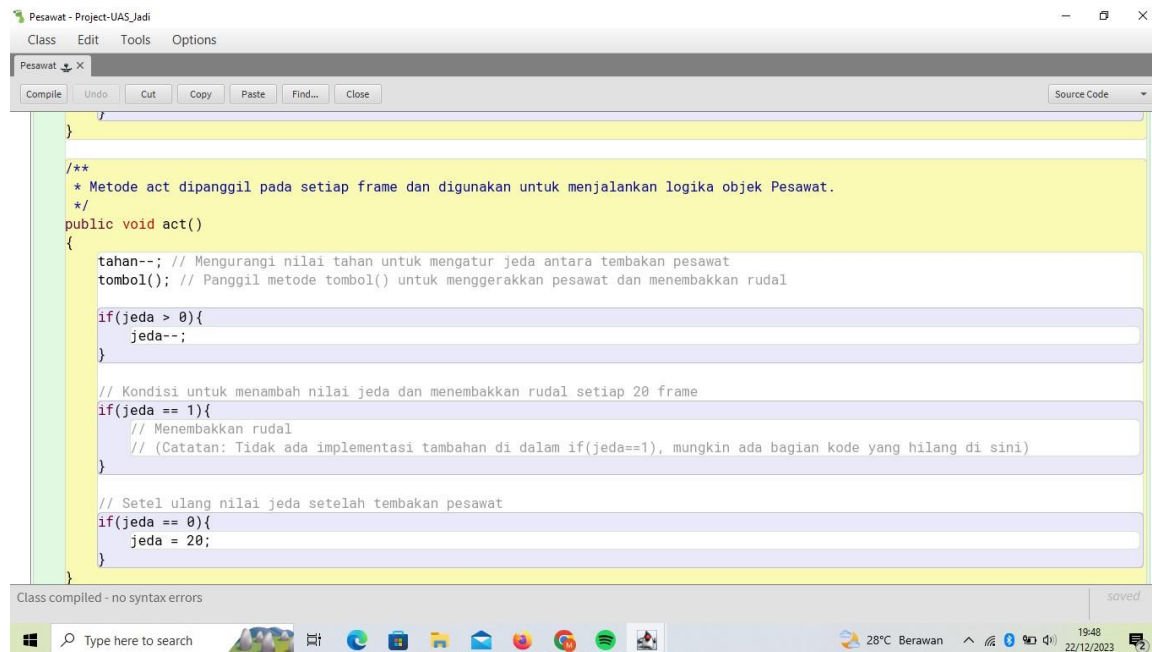
This screenshot shows the continuation of the Java code for the "Pesawat" class in the same IDE. The code continues from the previous block:

```
        if(Greenfoot.isKeyDown("left")){
            setLocation(getX() - 4, getY());
        }
        if(Greenfoot.isKeyDown("right")){
            setLocation(getX() + 4, getY());
        }

        // Menembakkan rudal jika jeda sudah habis
        if (tahan == 0){
            if(Greenfoot.isKeyDown("Space"))
            {
                getWorld().addObject(new Rudal(), getX(), getY());
                Greenfoot.playSound("bullet.wav");
            }
            tahan = 10; // Mengatur ulang nilai tahan setelah tembakan
        }
    }

    /**
     * Metode act dipanggil pada setiap frame dan digunakan untuk menjalankan logika objek Pesawat.
     */
    public void act()
    {
        tahan--; // Mengurangi nilai tahan untuk mengatur jeda antara tembakan pesawat
        tombol(); // Panggil metode tombol() untuk menggerakkan pesawat dan menembakkan rudal
    }
}
```

The IDE interface remains the same, with the status bar showing "Class compiled - no syntax errors" and "saved". The Windows taskbar at the bottom is also visible, showing the same system information as the previous screenshot.



```

Pesawat - Project-UAS_Jadi
Class Edit Tools Options

Pesawat
Compile Undo Cut Copy Paste Find... Close Source Code

}

/**
 * Metode act dipanggil pada setiap frame dan digunakan untuk menjalankan logika objek Pesawat.
 */
public void act()
{
    tahan--; // Mengurangi nilai tahan untuk mengatur jeda antara tembakan pesawat
    tombol(); // Panggil metode tombol() untuk menggerakkan pesawat dan menembakkan rudal

    if(jeda > 0){
        jeda--;
    }

    // Kondisi untuk menambah nilai jeda dan menembakkan rudal setiap 20 frame
    if(jeda == 1){
        // Menembakkan rudal
        // (Catatan: Tidak ada implementasi tambahan di dalam if(jeda==1), mungkin ada bagian kode yang hilang di sini)
    }

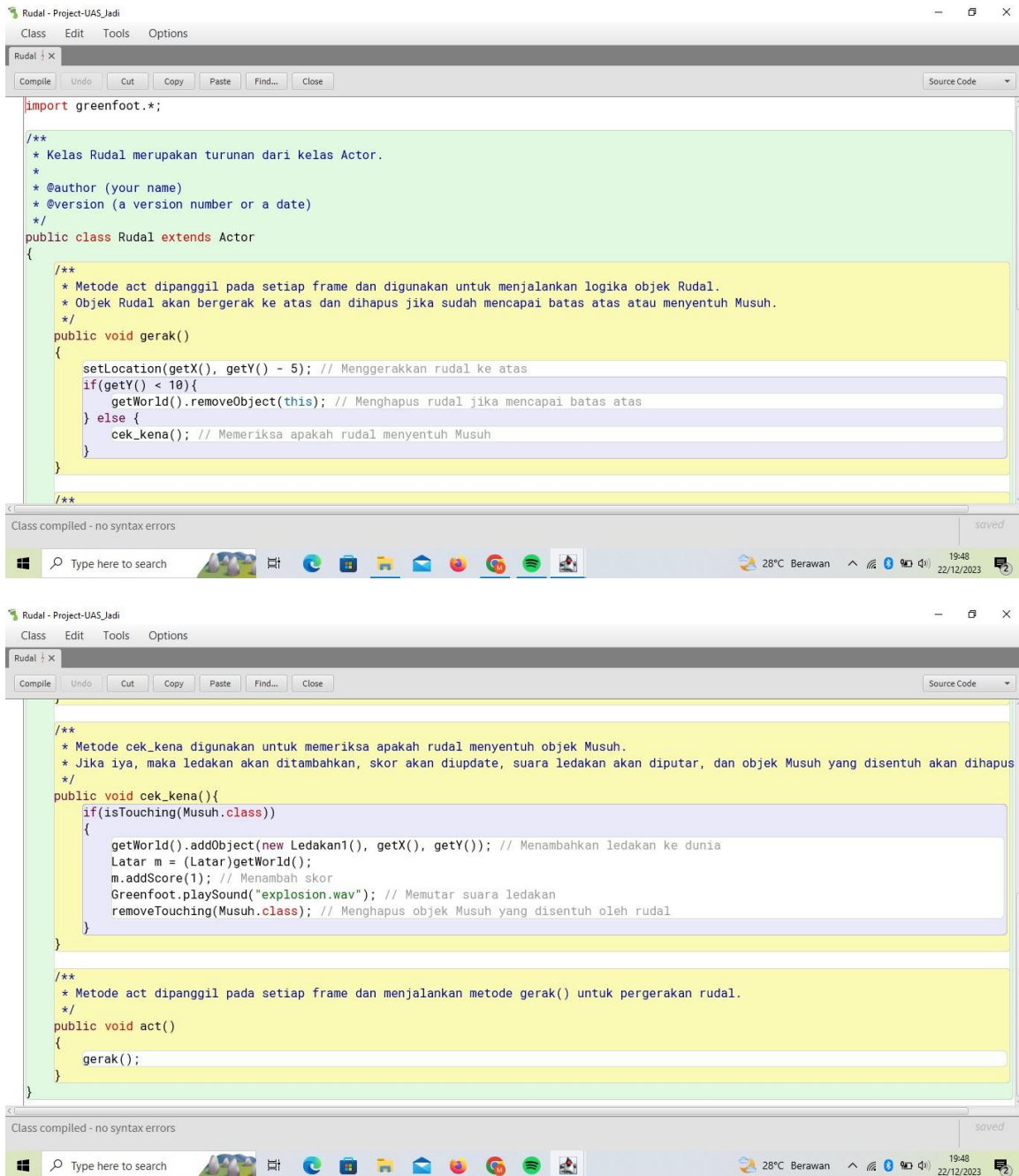
    // Setel ulang nilai jeda setelah tembakan pesawat
    if(jeda == 0){
        jeda = 20;
    }
}
}

Class compiled - no syntax errors
saved

```

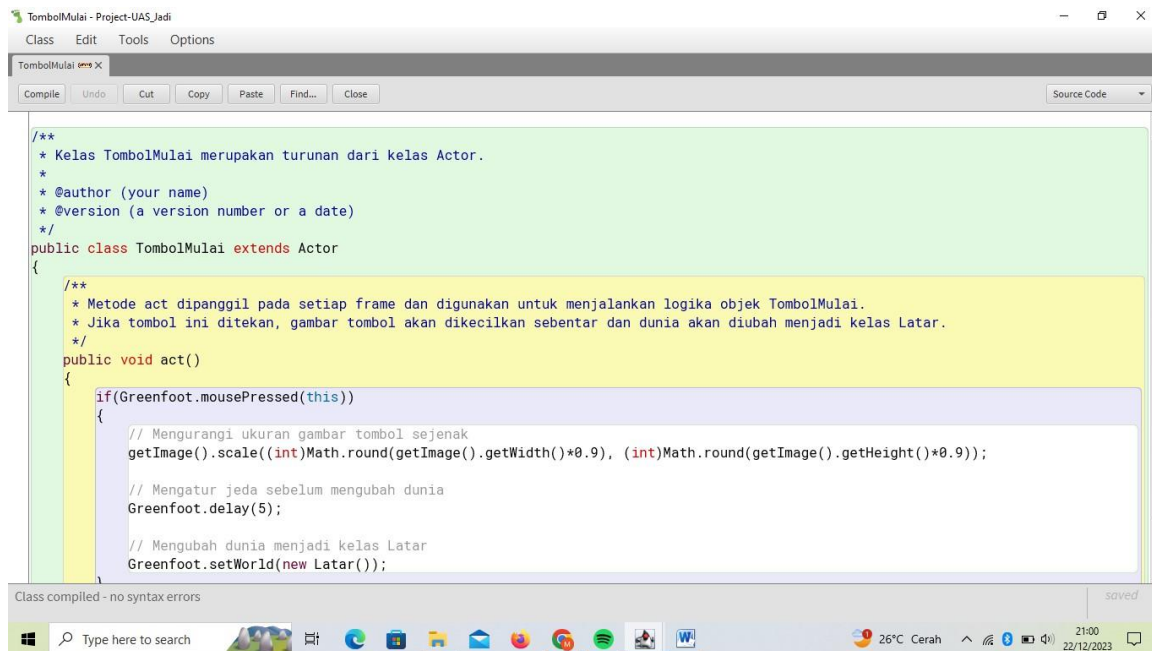
- Kelas Pesawat memiliki variabel tahan untuk mengatur jeda antara tembakan pesawat, jeda untuk mengatur waktu antara tembakan pesawat, dan UwU untuk mengatur nilai UwU dalam permainan. Metode tombol() menggerakkan pesawat berdasarkan input tombol kursor dan menembakkan rudal. Metode act() mengurangi nilai tahan, memanggil tombol(), dan menambah serta mengatur nilai jeda untuk menembakkan rudal setiap 20 frame. Perlu dicatat bahwa implementasi tembakan rudal mungkin belum lengkap atau ada bagian kode yang belum ditambahkan.

10. RUDAL



- Kelas Rudal merupakan turunan dari kelas Actor. Metode gerak() memindahkan rudal ke atas, dan jika mencapai batas atas, rudal akan dihapus. Metode juga memanggil cek_kenal() untuk memeriksa apakah rudal menyentuh objek Musuh. Jika ya, ledakan ditambahkan, skor diperbarui, suara ledakan diputar, dan objek Musuh yang disentuh dihapus. Metode act() memanggil gerak() untuk mengatur logika pergerakan rudal pada setiap frame.

11. TOMBOL MULAI



```
/**
 * Kelas TombolMulai merupakan turunan dari kelas Actor.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class TombolMulai extends Actor
{
    /**
     * Metode act dipanggil pada setiap frame dan digunakan untuk menjalankan logika objek TombolMulai.
     * Jika tombol ini ditekan, gambar tombol akan dikecilkan sebentar dan dunia akan diubah menjadi kelas Latar.
     */
    public void act()
    {
        if(Greenfoot.mousePressed(this))
        {
            // Mengurangi ukuran gambar tombol sejenak
            getImage().scale((int)Math.round(getImage().getWidth()*0.9), (int)Math.round(getImage().getHeight()*0.9));

            // Mengatur jeda sebelum mengubah dunia
            Greenfoot.delay(5);

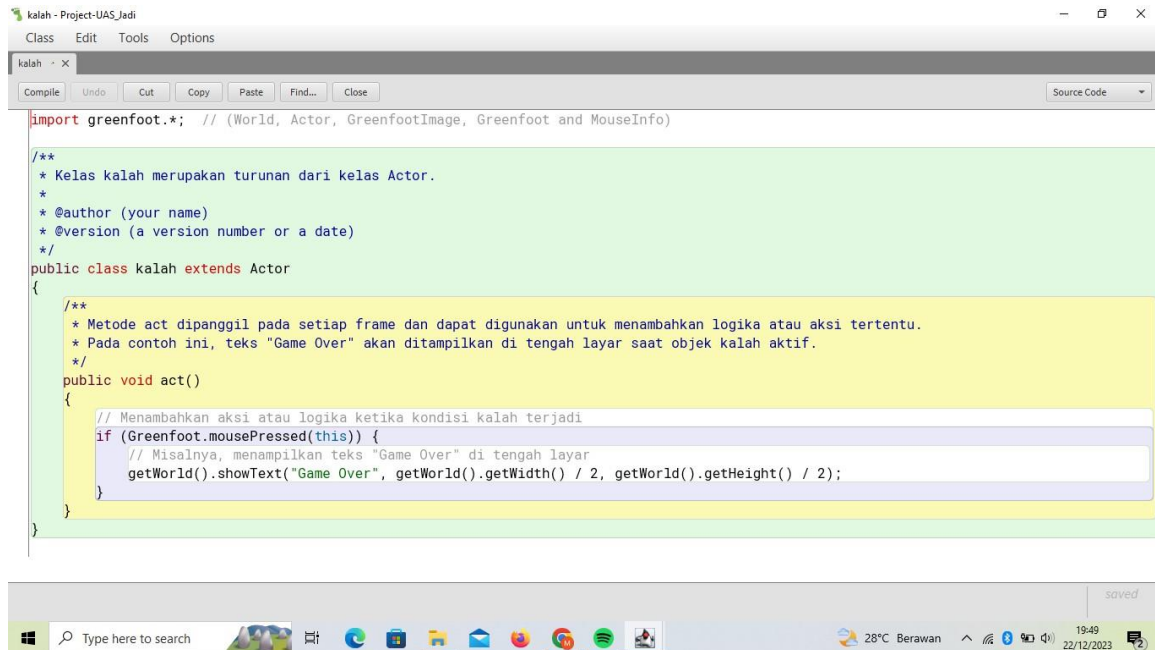
            // Mengubah dunia menjadi kelas Latar
            Greenfoot.setWorld(new Latar());
        }
    }
}
```

Class compiled - no syntax errors

21:00
22/12/2023

- Kelas TombolMulai merupakan turunan dari kelas Actor. Metode act() dipanggil pada setiap frame dan digunakan untuk menjalankan logika objek TombolMulai. Jika tombol ini ditekan (mousePressed), gambar tombol akan diperkecil sebentar, dan dunia akan diubah menjadi kelas Latar.

12. KALAH



- Kelas kalah merupakan turunan dari kelas Actor. Metode act() dipanggil pada setiap frame dan dapat digunakan untuk menambahkan logika atau aksi tertentu. Pada contoh ini, jika objek kalah ditekan (mousePressed), akan ditampilkan teks "Game Over" di tengah layar.

Kesimpulan:

“Kesimpulan dalam game ini adalah permainan Pesawat tempur sederhana dengan fokus pada kendali pesawat dan penghancuran musuh. Elemen-elemen seperti skor, nyawa (diwakili oleh "UwU"), serta kondisi kemenangan dan kekalahan memberikan struktur dan tujuan dalam permainan. Meskipun sederhana, game ini menawarkan pengalaman yang menghibur dengan tantangan untuk mencapai skor tertentu dengan diberikan 3 kesempatan nyawa. Game ini memiliki potensi untuk dikembangkan dengan menambahkan fitur-fitur baru, tingkat kesulitan yang beragam, atau bahkan elemen desain dan grafis yang lebih menarik sesuai keinginan.”