

Nama : Muhammad Haiqal Al Farizi

Kelas : 2TI03

NIM : 2222105142

Game : Tebak Angka

```
import javax.swing.*;

public class Player {
    private String name;

    public Player() {
        this.name = JOptionPane.showInputDialog("Masukkan nama pemain:");
        if (name == null || name.trim().isEmpty()) {
            this.name = "Pemain";
        }
    }

    public String getName() {
        return name;
    }
}
```

Syntax di atas adalah bagian dari sebuah program Java yang mendefinisikan sebuah kelas bernama Player. Berikut ini adalah penjelasan secara detail dari setiap bagian syntax yang terdapat dalam program tersebut:

1. Import Statement:

- `import javax.swing.*;`

Ini adalah pernyataan impor yang mengimpor keseluruhan paket `javax.swing`, yang merupakan paket Java untuk antarmuka grafis (GUI). `JOptionPane` yang digunakan nantinya adalah bagian dari paket ini.

2. Deklarasi Kelas 'Player':

- `public class Player {`

Deklarasi kelas dimulai dengan kata kunci `public` yang menunjukkan bahwa kelas ini dapat diakses dari luar paket (`public`). Nama kelasnya adalah 'Player'.

3. Variabel Instance 'name':

- `private String name;`

Variabel instance 'name' bertipe 'String' yang bersifat private. Ini berarti variabel 'name' ini hanya dapat diakses langsung dari dalam kelas 'Player' ini saja.

4. Konstruktor Kelas 'Player':

- ```
public Player() {
 this.name = JOptionPane.showInputDialog("Masukkan nama pemain:");
 if (name == null || name.trim().isEmpty()) {
 this.name = "Pemain";
 }
}
```

Konstruktor ini merupakan metode khusus yang dipanggil saat objek dari kelas 'Player' dibuat. Pada konstruksi objek, konstruktor ini akan menampilkan dialog input menggunakan 'JOptionPane.showInputDialog'. Dialog ini meminta pengguna untuk memasukkan nama pemain.

- a. 'JOptionPane.showInputDialog'("Masukkan nama pemain:"); digunakan untuk menampilkan kotak dialog input di GUI yang meminta pengguna memasukkan nama pemain.
- b. Setelah nama dimasukkan, nilai yang dimasukkan oleh pengguna disimpan dalam variabel 'name' menggunakan pernyataan 'this.name = ...;'
- c. Dilakukan pemeriksaan apakah input nama null atau kosong menggunakan 'if (name == null || name.trim().isEmpty())'. Jika ya, maka nilai 'this.name' diatur menjadi "Pemain" sebagai nilai default.

### 5. Metode getName():

- ```
public String getName() {  
    return name;  
}
```

Metode 'getName()' digunakan untuk mengembalikan nilai dari variabel 'name'. Ini adalah metode getter yang memungkinkan penggunaan publik untuk mendapatkan nilai 'name' dari objek 'Player'.

Keseluruhan kelas 'Player' ini dirancang untuk mengizinkan pembuatan objek 'Player' dengan nama yang dimasukkan pengguna melalui dialog input, dengan nilai default "Pemain" jika nama tidak dimasukkan atau inputnya kosong. Metode 'getName()' digunakan untuk mendapatkan nama pemain setelah objek 'Player' dibuat.

```
public class Main {
    public static void main(String[] args) {
        Game game = new Game();
        game.start();
    }
}
```

Syntax di atas adalah bagian dari sebuah program Java yang terdiri dari sebuah kelas utama ('Main') dengan sebuah method 'main'. Mari kita jabarkan detailnya:

1. Deklarasi Kelas Main:

- `public class Main {`
`// Isi dari kelas Main akan ada di sini`
`}`

'`public class Main { ... }`': Mendefinisikan kelas dengan nama 'Main'. Kata kunci `public` menandakan bahwa kelas ini dapat diakses dari kelas lain di luar paket ('package').

2. Method main:

- `public static void main(String[] args) {`
`// Isi dari method main akan ada di sini`
`}`
 - '`public static void main(String[] args) { ... }`': Ini adalah method utama dalam sebuah program Java. Method ini akan dijalankan pertama kali ketika program Java dijalankan.
 - '`public`': Kata kunci `public` menandakan bahwa method `main` dapat diakses dari luar kelas 'Main' (yaitu oleh JVM ketika menjalankan program).
 - '`static`': Kata kunci `static` menandakan bahwa method 'main' adalah method kelas (bukan method objek), dan oleh karena itu bisa dipanggil tanpa harus membuat objek dari kelas 'Main'.
 - '`void`': Kata kunci '`void`' menunjukkan bahwa method 'main' tidak mengembalikan nilai apapun.
 - '`main`': Nama method ini adalah 'main', yang merupakan nama yang telah ditentukan secara khusus untuk method utama dalam program Java.
 - '`String[] args`': Parameter dari method 'main' adalah array dari `String` yang dapat digunakan untuk menerima argumen baris perintah saat program dijalankan.

3. Membuat Objek Game dan Memanggil Method start():

- `Game game = new Game();`
`game.start();`

`'Game game = new Game();'` : Membuat sebuah objek baru dari kelas `'Game'` dengan menggunakan operator `'new'`. Objek ini disimpan dalam variabel `'game'`.

`'game.start();'` : Memanggil method `'start()'` dari objek `'game'`. Method ini akan memulai eksekusi atau menjalankan logika yang terkait dengan permainan (`'Game'`).

Dengan demikian, kode di atas akan membuat sebuah objek dari kelas `'Game'` dan memanggil method `'start()'` dari objek tersebut, yang kemungkinan besar akan menginisialisasi dan menjalankan logika utama dari permainan yang dibuat dalam kelas `'Game'`.

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.Random;

public class Game {
    private Player player;
    private int targetNumber;
    private int maxGuesses;
    private int currentGuesses;
    private JFrame frame;
    private JTextField guessField;
    private JLabel feedbackLabel;
    private JLabel guessesLabel;

    public Game() {
        this.player = new Player();
        this.targetNumber = generateTargetNumber();
        this.maxGuesses = 10;
        this.currentGuesses = 0;
        setupGUI();
    }

    public void start() {
        frame.setVisible(true);
    }

    private void setupGUI() {
        frame = new JFrame("Tebak Angka");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```

        frame.setSize(300, 200);
        frame.setLayout(new GridLayout(5, 1));

        JLabel welcomeLabel = new JLabel("Selamat datang di game tebak angka, " + player.getName() + "!");
        frame.add(welcomeLabel);

        guessField = new JTextField();
        frame.add(guessField);

        JButton guessButton = new JButton("Tebak");
        frame.add(guessButton);

        feedbackLabel = new JLabel("Masukkan angka antara 1 dan 100");
        frame.add(feedbackLabel);

        guessesLabel = new JLabel("Tebakan tersisa: " + (maxGuesses - currentGuesses));
        frame.add(guessesLabel);

        guessButton.addActionListener(new GuessButtonListener());
    }

    private class GuessButtonListener implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent e) {
            String guessText = guessField.getText();
            try {
                int guess = Integer.parseInt(guessText);
                currentGuesses++;

                if (guess == targetNumber) {
                    feedbackLabel.setText("Selamat, " + player.getName() + "! Anda berhasil menebak angka!");
                    endGame();
                } else if (guess < targetNumber) {
                    feedbackLabel.setText("Tebakan Anda terlalu rendah.");
                } else {
                    feedbackLabel.setText("Tebakan Anda terlalu tinggi.");
                }

                if (currentGuesses >= maxGuesses) {
                    feedbackLabel.setText("Maaf, " + player.getName() + ". Anda kehabisan tebak. Angka yang benar adalah " + targetNumber);
                    endGame();
                }
            }
        }
    }

```

```

        guessesLabel.setText("Tebakan tersisa: " + (maxGuesses -
currentGuesses));
    } catch (NumberFormatException ex) {
        feedbackLabel.setText("Masukkan angka yang valid.");
    }
}

private void endGame() {
    guessField.setEnabled(false);
}

private int generateTargetNumber() {
    Random random = new Random();
    return random.nextInt(100) + 1;
}
}

```

Syntax di atas adalah bagian dari sebuah program Java yang mengimplementasikan permainan tebak angka dengan antarmuka grafis menggunakan ‘Java Swing’. Mari kita jabarkan detailnya:

1. Imports:

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.Random;

```

- o `javax.swing.*`: Package ini menyediakan kelas-kelas untuk membuat GUI dalam Java.
- o `java.awt.*`: Package ini menyediakan kelas-kelas dasar untuk mengelola antarmuka pengguna dalam Java.
- o `java.awt.event.ActionEvent` dan `java.awt.event.ActionListener`: Keduanya digunakan untuk menangani event (peristiwa) pada komponen GUI seperti tombol.
- o `java.util.Random`: Digunakan untuk menghasilkan angka acak.

2. Deklarasi Kelas Game:

```

public class Game {
    // Deklarasi variabel instance
    private Player player;
    private int targetNumber;
    private int maxGuesses;
    private int currentGuesses;
    private JFrame frame;
    private JTextField guessField;

```

```

private JLabel feedbackLabel;
private JLabel guessesLabel;

// Konstruktor
public Game() {
    // Inisialisasi variabel instance
    this.player = new Player();
    this.targetNumber = generateTargetNumber();
    this.maxGuesses = 10;
    this.currentGuesses = 0;
    setupGUI(); // Setup GUI ketika objek Game dibuat
}

// Method untuk memulai permainan
public void start() {
    frame.setVisible(true); // Menampilkan frame GUI
}

// Metode untuk mengatur antarmuka pengguna
private void setupGUI() {
    // Membuat frame utama
    frame = new JFrame("Tebak Angka");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); //
    Menutup aplikasi jika frame ditutup
    frame.setSize(300, 200); // Menentukan ukuran frame
    frame.setLayout(new GridLayout(5, 1)); // Mengatur layout
    frame menggunakan GridLayout 5 baris, 1 kolom

    // Menambahkan komponen-komponen GUI ke dalam frame
    JLabel welcomeLabel = new JLabel("Selamat datang di game
    tebak angka, " + player.getName() + "!");
    frame.add(welcomeLabel); // Menambahkan label selamat datang

    guessField = new JTextField(); // Membuat field untuk menebak
    angka
    frame.add(guessField); // Menambahkan field tebakan ke dalam
    frame

    JButton guessButton = new JButton("Tebak"); // Membuat tombol
    tebak
    frame.add(guessButton); // Menambahkan tombol tebak ke dalam
    frame

    feedbackLabel = new JLabel("Masukkan angka antara 1 dan
    100"); // Label untuk memberi umpan balik
    frame.add(feedbackLabel); // Menambahkan label umpan balik ke
    dalam frame

    guessesLabel = new JLabel("Tebakan tersisa: " + (maxGuesses -
    currentGuesses)); // Label untuk menampilkan tebakan tersisa
    frame.add(guessesLabel); // Menambahkan label tebakan tersisa
    ke dalam frame

    guessButton.addActionListener(new GuessButtonListener()); //
    Menambahkan listener untuk tombol tebak
}

// Kelas inner private untuk mengimplementasikan ActionListener
untuk tombol tebak
private class GuessButtonListener implements ActionListener {
    @Override

```

```

        public void actionPerformed(ActionEvent e) {
            String guessText = guessField.getText(); // Mendapatkan
            teks dari field tebakan
            try {
                int guess = Integer.parseInt(guessText); // Mengubah
                teks menjadi integer
                currentGuesses++; // Menambah jumlah tebakan yang
                sudah dilakukan

                // Memeriksa apakah tebakan benar, terlalu rendah,
                atau terlalu tinggi
                if (guess == targetNumber) {
                    feedbackLabel.setText("Selamat, " +
                    player.getName() + "! Anda berhasil menebak angka!");
                    endGame(); // Memanggil method endGame karena
                    permainan selesai
                } else if (guess < targetNumber) {
                    feedbackLabel.setText("Tebakan Anda terlalu
                    rendah.");
                } else {
                    feedbackLabel.setText("Tebakan Anda terlalu
                    tinggi.");
                }

                // Memeriksa apakah sudah melebihi jumlah maksimal
                tebakan
                if (currentGuesses >= maxGuesses) {
                    feedbackLabel.setText("Maaf, " + player.getName()
                    + ". Anda kehabisan tebakan. Angka yang benar adalah " +
                    targetNumber);
                    endGame(); // Memanggil method endGame karena
                    permainan selesai
                }

                guessesLabel.setText("Tebakan tersisa: " +
                (maxGuesses - currentGuesses)); // Memperbarui label tebakan tersisa
            } catch (NumberFormatException ex) {
                feedbackLabel.setText("Masukkan angka yang valid.");
            }
            // Menangani exception jika input bukan angka
        }

        // Method untuk mengakhiri permainan
        private void endGame() {
            guessField.setEnabled(false); // Menonaktifkan field tebakan
            agar tidak bisa diisi lagi
        }

        // Method untuk menghasilkan angka target secara acak
        private int generateTargetNumber() {
            Random random = new Random();
            return random.nextInt(100) + 1; // Menghasilkan angka acak
            antara 1 sampai 100
        }
    }

```


Dalam kode di atas, kita memiliki:

- **Deklarasi Variabel:** Berbagai variabel instance seperti `player`, `targetNumber`, `maxGuesses`, `currentGuesses`, `frame`, `guessField`, `feedbackLabel`, dan `guessesLabel` untuk menyimpan informasi tentang permainan dan antarmuka pengguna.
- **Konstruktur `Game()`:** Menginisialisasi variabel instance dan memanggil `setupGUI()` untuk mengatur antarmuka pengguna.
- **Method `start()`:** Menampilkan frame GUI ketika permainan dimulai.
- **Method `setupGUI()`:** Membuat frame GUI dengan menggunakan `JFrame`, `JLabel`, `JTextField`, dan `JBUTTON` untuk memungkinkan interaksi pengguna.
- **Private Inner Class `GuessButtonListener`:** Mengimplementasikan `ActionListener` untuk tombol tebakan (`JBUTTON guessButton`). Ketika tombol ditekan, method `actionPerformed()` akan dieksekusi untuk memeriksa tebakan pengguna dan memberikan umpan balik.
- **Method `endGame()`:** Menonaktifkan field tebakan setelah permainan selesai.
- **Method `generateTargetNumber()`:** Menghasilkan angka acak sebagai angka yang harus ditebak oleh pemain.

Kode ini secara keseluruhan menggabungkan penggunaan Java Swing untuk membuat antarmuka pengguna GUI sederhana yang memungkinkan pengguna untuk berinteraksi dengan permainan tebak angka.