Nama & NPM	Topik:	Tanggal:
RIZQI NADHIFAH SETIANI G1F024017	Kelas (Class)	**18/09/2024**

[No.1] Identifikasi Masalah:

```
public class Manusia {  // deklarasi kelas
    //deklarasi atribut Manusia dalam variabel
    String nama, rambut;

    //deklarasi constructor
    public Manusia1 (String nama) {
            System.out.println(" Nama saya : "+ nama +
            "\n Warna Rambut : " + rambut);
    }

    //deklarasi method utama
    public static void main( String[] args) {
            Manusia1 satu = new Manusia1("Putri", "hitam");
    }
  }
```

Luaran 1:

```
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
    The constructor Manusia1(String, String) is undefined
    at Manusia1.main(Manusia1.java:13)
```

Pada kode pemroraman di atas, kesalahan pada kode terjadi karena Nama Class dan Constructor Tidak Konsisten: Nama class adalah Manusia, tetapi constructor yang dideklarasikan adalah Manusia1. Ini menyebabkan error karena tidak ada constructor Manusia1 yang sesuai dengan class Manusia. Kemudian, Jumlah Parameter Constructor: Constructor Manusia1(String nama) hanya memiliki satu parameter (nama), sedangkan saat memanggilnya, dua parameter ("Putri", "hitam") diberikan. Jadi, constructor yang benar harus menerima dua parameter: String nama dan String rambut.

[No.1] Analisis dan Argumentasi

Latihan 1:

1.1. Perbaiki pesan kesalahan Contoh 1!

1.2. Analisa ciri-ciri lain Kelas Manusia yang dapat menjadi
   a. atribut variabel, dan
   b. perilaku/ behavior!

Perbaikan pesan kesalahan pada contoh 1 dapat dilakukan dengan membuat Nama Class dan Constructor yang konsisten: Constructor Manusia harus sesuai dengan nama class, jadi diubah dari Manusia1 ke Manusia. Kemudian constructor menerima dua parameter: String nama dan String rambut, sesuai dengan pemanggilan constructor pada method main.

Atribut variabel lain yang dapat ditambahkan untuk class Manusia mencakup karakteristik fisik atau sifat umum yang dimiliki manusia di antaranya : umur, jenis kelamin, dan tinggi badan. Sementara perilaku manusia bisa berupa aktivitas yang dilakukan, seperti berjalan dan berbicara.

[No.1] Penyusunan Algoritma dan Kode Program

1) Rancang desain solusi atau algoritma
   - Mulai program eclipse
   - Deklarasi program “Manusia”
   - Lakukan perbaikan pada soal

- Deklarasi atribut umur, jenis kelamin, dan tinggi badan.
- Definisikan constructor dengan parameter untuk menginisialisasi atribut.
- Deklarasi Method Perilaku
- Jalankan Method

2) Tuliskan kode program dan luaran

```java
package Rizqi.Nadhifah;
public class Manusia {  // deklarasi kelas
    // deklarasi atribut Manusia dalam variabel
    String nama, rambut;
    int umur;
    String jenisKelamin;
    double tinggiBadan;
    // deklarasi constructor
    public Manusia(String nama, String rambut, int umur, String jenisKelamin, double tinggiBadan) {
        this.nama = nama;
        this.rambut = rambut;
        this.umur = umur;
        this.jenisKelamin = jenisKelamin;
        this.tinggiBadan = tinggiBadan;

        System.out.println("Nama saya: " + nama +
                           "\nWarna Rambut: " + rambut +
                           "\nUmur: " + umur +
                           "\nJenis Kelamin: " + jenisKelamin +
                           "\nTinggi Badan: " + tinggiBadan );
    }

    // deklarasi method
    public void berjalan() {
        System.out.println(nama + " sedang berjalan.");
    }

    public void berbicara(String kata) {
        System.out.println(nama + " berkata: " + kata);
    }

    // deklarasi method utama
    public static void main(String[] args) {
        Manusia satu = new Manusia("Kiki", "hitam", 18, "Perempuan", 159.5);
        satu.berjalan();
        satu.berbicara("Halo, apa kabar?");
    }
```

```
Nama saya: Kiki
Warna Rambut: hitam
Umur: 18
Jenis Kelamin: Perempuan
Tinggi Badan: 159.5
Kiki sedang berjalan.
Kiki berkata: Halo, apa kabar?
```

Luaran yang dihasilkan telah sesuai.

[No.1] Kesimpulan

Atribut adalah variabel yang menggambarkan karakteristik atau data dari objek manusia. Pada kode program tersebut ada beberapa atribut, yaitu:
nama: Menyimpan nama manusia (tipe String).
rambut: Menyimpan warna rambut manusia (tipe String).
jenisKelamin: Menyimpan jenis kelamin manusia (tipe String).
umur: Menyimpan umur manusia (tipe int, untuk bilangan bulat).
tinggi: Menyimpan tinggi badan manusia (tipe double, untuk angka desimal).
Parameter constructor (seperti String nama, String rambut, dll.) adalah nilai yang kita masukkan saat membuat objek. Nilai ini kemudian disimpan ke dalam atribut objek menggunakan this.nama = nama dan seterusnya. Constructor ini juga menampilkan informasi tentang manusia yang baru dibuat dengan menggunakan System.out.println(). Kemudian, Method adalah fungsi yang menggambarkan perilaku atau aksi yang dapat dilakukan oleh objek. Setiap method ini hanya mencetak aktivitas yang dilakukan oleh manusia tersebut. void menunjukkan bahwa method ini tidak mengembalikan nilai. Saat objek Manusia dibuat, constructor akan menampilkan informasi

atribut dari objek satu dan program akan menampilkan hasil dari method.

Nama & NPM	Topik:	Tanggal:
RIZQI NADHIFAH SETIANI G1F024017	Objek	**18/09/2024**

[No.2] Identifikasi Masalah:

```
public class Ortu {
    //deklarasi constructor  (variabel constructor)
    public ortu   {
            //nama dan rambut adalah variabel constructor
         System.out.println(" Nama saya : "+ nama +
        "\n Warna Rambut : " + rambut);
}
    public static void main (String[] args) {
        Ortu satu = new Ortu("Putri", "hitam");
     }
}
```

Luaran 2:

```
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
    The constructor Ortu(String, String) is undefined

    at Ortu.main(Ortu.java:9)
```

Pada kode pemroraman di atas, kesalahan pada kode terjadi pada constructor nya. Di dalam Java, constructor harus memiliki nama yang sama dengan nama kelas. Dalam kode soal, nama class adalah Ortu, tetapi pada deklarasi constructor, nama ortu (huruf kecil) digunakan. Ini menyebabkan error karena Java tidak bisa mengenali constructor yang tidak sesuai namanya dengan class. Selain itu, parameter constructor belum dideklarasikan dengan benar. Constructor harus menerima parameter nama dan rambut agar bisa digunakan dalam System.out.println.

[No.2] Analisis dan Argumentasi

Latihan 2:

2.1. Evaluasi penyebab kesalahan dan perbaiki kode tersebut!
2.2. Apabila Ortu memiliki data variabel umur = 25 dan jenis kelamin = P (untuk Perempuan), rekomendasikan constructor dengan parameter yang baru untuk ditambahkan dalam program!

Perbaikan pesan kesalahan pada soal ini dapat dilakukan dengan mengubah nama constructor dari ortu menjadi Ortu agar sesuai dengan nama class. Lalu tambahkan parameter pada constructor untuk menerima nilai nama dan rambut. Dengan Constructor Ortu(String nama, String rambut)maka Constructor ini sekarang memiliki parameter nama dan rambut, dan pada variabel this.nama serta this.rambut menyimpan nilai dari parameter tersebut. Method main(), objek Ortu satu dibuat dengan mengirimkan dua nilai ("Putri" dan "hitam") ke constructor.
Untuk menambahkan umur dan jenis kelamin sebagai parameter baru dalam constructor, tambahkan dua atribut baru, umur dan jenisKelamin, ke class Ortu. Inisialisasi atribut tersebut dalam constructor dan tambahkan output untuk menampilkan informasi ini. Constructor sekarang menerima 4 parameter: nama, rambut, umur, dan jenisKelamin, yang digunakan untuk menginisialisasi atribut yang sesuai.

[No.2] Penyusunan Algoritma dan Kode Program

Rancang desain solusi atau algoritma

- Mulai program eclipse
- Definisikan sebuah class bernama Ortu
- Lakukan perbaikan pada soal
- Deklarasi atribut umur dan jenis kelamin
- Definisikan constructor dengan parameter untuk menginisialisasi atribut.
- Deklarasi Method Perilaku

- Jalankan program dan cetak informasi atribut dari objek satu.

Tuliskan kode program dan luaran

```java
package Rizqi.Nadhifah;

public class Ortu {
    // deklarasi atribut
    String nama, rambut;
    int umur;
    String jenisKelamin;

    // deklarasi constructor
    public Ortu(String nama, String rambut, int umur, String jenisKelamin) {
        this.nama = nama;
        this.rambut = rambut;
        this.umur = umur;
        this.jenisKelamin = jenisKelamin;
        System.out.println("Nama saya: " + nama +
                           "\nWarna Rambut: " + rambut +
                           "\nUmur: " + umur +
                           "\nJenis Kelamin: " + jenisKelamin);
    }

    public static void main(String[] args) {
        Ortu satu = new Ortu("Putri", "hitam", 25, "P");
    }
}

```

Problems Javadoc Declaration Console

```
<terminated> Ortu [Java Application] C:\Program Files\Java\jre1.8.0_421\bin\javaw.exe (Sep 21, 2024, 7:34:07 AM
Nama saya: Putri
Warna Rambut: hitam
Umur: 25
Jenis Kelamin: P
```

Luaran yang dihasilkan telah sesuai.

[No.2] Kesimpulan

Pada kode soal ini perbaikan error dilakukan karena nama constructor harus sesuai dengan nama class (Ortu).Penting untuk mendeklarasikan constructor dengan parameter yang tepat agar program bisa berjalan tanpa kesalahan, Penambahan parameter (seperti umur dan jenisKelamin) memberikan fleksibilitas pada class Ortu untuk menyimpan lebih banyak informasi mengenai objek yang diciptakan, memperluas kemampuan class tersebut.

Nama & NPM	Topik:	Tanggal:
RIZQI NADHIFAH SETIANI G1F024017	Method	**18/09/2024**

[No.3] Identifikasi Masalah:

```
public class Manusia {
    //deklarasi atribut Manusia dalam variabel
    String nama, rambut;

    //deklarasi constructor
    public Manusia1(String nama, String rambut) {
            System.out.println(" Nama saya : "+ nama +
            "\n Warna Rambut : " + rambut);
    }

    //deklarasi method
    void sukaNonton  {
        System.out.println(" Hobi Menonton : " + film);
    }

    int sukaNonton {
        episode*durasi;
    }

    //deklarasi method utama
    public static void main( String[] args) {
            Manusia satu = new Manusia("Putri", "hitam");
            satu.sukaNonton("Drakor");
            int jumlahJam = satu.sukaNonton(2, 2);
            System.out.println("Jam nonton = " +jumlahJam + " jam");
    }
}
```

Luaran 3:

```
Exception in thread "main" java.lang.Error: Unresolved compilation problems:
    The method sukaNonton(String) is undefined for the type Manusia1
    The method sukaNonton(int, int) is undefined for the type Manusia1
    at Manusia1.main(Manusia1.java:23)
```

Pada kode pemroraman di atas, beberapa kesalahan yang ditemukan dalam kode yaitu nama class adalah Manusia, tetapi constructor diberi nama Manusia1. Ini menyebabkan error karena constructor seharusnya dinamai Manusia. Kemudian method sukaNonton tidak dideklarasikan dengan benar. Sintaks method harus menggunakan tanda kurung buka () setelah nama method untuk menandakan bahwa itu adalah method, misalnya void sukaNonton(). Lalu method sukaNonton(String film) tidak ada, sementara dipanggil di main(). Ini menyebabkan error karena Java tidak dapat menemukan method tersebut.Yang terakhir, ada method kedua dengan nama sukaNonton(int episode, int durasi), tetapi tidak dideklarasikan dengan benar dan operator matematis di dalamnya salah.

[No.3] Analisis dan Argumentasi

Latihan 3:

3.1. Evaluasi penyebab kesalahan dan perbaiki kode tersebut!
3.2. Ubahlah method dan constructor Contoh 3 sesuai dengan perilaku/ behavior anda
3.3. Berdasarkan Contoh 3 dan Latihan 3.2. simpulkan perbedaan:
    a) constructor overloading dan overriding

b) method overloading, dan method overriding
c) method yang mengembalikan nilai dan method tidak mengembalikan nilai

Perbaikan pesan kesalahan pada soal ini dapat dilakukan dengan memperbaiki constructor menggunakan nama yang sama dengan class (Manusia), dan menerima parameter nama dan rambut untuk menginisialisasi atribut. Method sukaNonton(String film)akan menampilkan film yang sedang ditonton, sesuai dengan input parameter film. Method sukaNonton(int episode, int durasi)akan mengembalikan hasil dari perkalian jumlah episode dengan durasi per episode untuk menghitung total jam menonton.

Kemudian untuk mengubah method dan constructor Contoh 3 sesuai dengan perilaku/ behavior, saya menggunakan Method sukaMembaca. Method ini menerima parameter berupa String buku dan mencetak hobi membaca buku yang diberikan. Ini adalah contoh dari method overloading karena ada method lain dengan nama yang sama tetapi parameter yang berbeda. Method ini menerima dua parameter (int halaman dan int waktuPerHalaman) dan mengembalikan total waktu yang diperlukan untuk membaca sejumlah halaman, dihitung dengan mengalikan halaman dengan waktuPerHalaman. Method sukaMembaca(String buku) dipanggil dengan parameter "Komik", mencetak hobi membaca. Method sukaMembaca(int halaman, int waktuPerHalaman) dipanggil dengan nilai 20 untuk halaman dan 5 untuk waktu per halaman, menghitung total waktu membaca dan menyimpannya dalam variabel jumlahWaktu.

Dapat disimpulkan perbedaan :

Constructor Overloading terjadi ketika ada lebih dari satu constructor dengan nama yang sama, tetapi dengan parameter yang berbeda. Misalnya, constructor Manusia(String nama, String rambut) bisa di-overload dengan constructor lain yang menerima parameter tambahan seperti Manusia(String nama, String rambut, int umur). **Constructor Overriding t**idak berlaku karena constructor tidak bisa di-override (hanya bisa di-overload). **Method Overloading** dapat terjadi ketika dua atau lebih method memiliki nama yang sama tetapi parameter yang berbeda. Dalam contoh ini, method sukaNonton(String film) di-overload dengan sukaNonton(int episode, int durasi).**Method Overriding** akan terjadi ketika subclass mendefinisikan ulang method dari superclass dengan nama, parameter, dan tipe return yang sama. Ini memungkinkan subclass memiliki perilaku method yang berbeda dari superclass. **Method yang Mengembalikan Nilai**: Method yang mengembalikan hasil, seperti sukaNonton(int episode, int durasi), mengembalikan nilai integer hasil perkalian. **Method yang Tidak Mengembalikan Nilai**: Method yang hanya melakukan tugas, seperti sukaNonton(String film), hanya mencetak sesuatu tanpa mengembalikan nilai (void).

[No.3] Penyusunan Algoritma dan Kode Program

Rancang desain solusi atau algoritma

- Mulai program eclipse
- Definisikan sebuah class bernama Manusia
- Lakukan perbaikan pada soal
  Deklarasi Method untuk Hobi Membaca (Overloading)
- Deklarasi Method untuk Menghitung Waktu Membaca (Overloading)
- Deklarasi Method main
- Jalankan program

Tuliskan kode program dan luaran

```java
package Rizqi.Nadhifah;
public class Manusia {
    String nama, rambut;

    public Manusia(String nama, String rambut) {
        this.nama = nama;
        this.rambut = rambut;
        System.out.println(" Nama saya : " + nama +
            "\n Warna Rambut : " + rambut);
    }

    void sukaMembaca(String buku) {
        System.out.println(" Hobi Membaca : " + buku);
    }

    int sukaMembaca(int halaman, int waktuPerHalaman) {
        return halaman * waktuPerHalaman;
    }

    public static void main(String[] args) {
        Manusia satu = new Manusia("Kiki", "hitam");
        satu.sukaMembaca("Komik");
        int jumlahWaktu = satu.sukaMembaca(20, 5);
        System.out.println("Waktu membaca = " + jumlahWaktu + " menit");
    }
}
```

Problems Javadoc Declaration Console

<terminated> Manusia [Java Application] C:\Program Files\Java\jre1.8.0_421\bin\javaw.exe (Sep 21, 2024,

```
 Nama saya : Kiki
 Warna Rambut : hitam
 Hobi Membaca : Komik
Waktu membaca = 100 menit
```

Luaran yang dihasilkan telah sesuai.

[No.3] Kesimpulan

Pada praktikum ini, objek Manusia bernama satu dibuat dengan memberikan nilai "Kiki" untuk nama dan "hitam" untuk rambut. Constructor Manusia dipanggil, yang mencetak nama dan warna rambut. Method sukaMembaca(String buku) dipanggil dengan parameter "Komik", mencetak hobi membaca. Method sukaMembaca(int halaman, int waktuPerHalaman) dipanggil dengan nilai 20 untuk halaman dan 5 untuk waktu per halaman, menghitung total waktu membaca dan menyimpannya dalam variabel jumlahWaktu. Praktikum ini menunjukkan bagaimana method dengan nama yang sama dapat memiliki parameter yang berbeda (overloading). Ini memungkinkan fleksibilitas dalam penggunaan method, seperti menghitung waktu membaca berdasarkan jumlah halaman dan waktu per halaman.

Nama & NPM	Topik:	Tanggal:
RIZQI NADHIFAH SETIANI G1F024017	Extends	**18/09/2024**

[No.4] Identifikasi Masalah:

```
public class Ortu {       // membuat kelas induk
  void sukaMenonton(String a) {    // method induk spesifik
    System.out.println("Nonton " + a);
  }
  void sukaMembaca(String a) {     // method induk umum bisa diubah anak
    System.out.println("Suka Baca " + a);
  }

public static void main(String [] args) {
    System.out.println("Sifat Orang Tua :");
    Ortu objekO = new Ortu();     // memanggil objek induk
    objekO.sukaMenonton("Berita");    // memanggil sifat spesifik induk
    objekO.sukaMembaca("Koran");     // memanggil method dengan variabel dapat
diubah

    System.out.println("\n Sifat Anak :");
    Anak objekA = new Anak();    //memanggil objek anak
    objekA.sukaMenonton(9, "Film Drakor");        //memanggil sifat spesifik anak
yang diturunkan induk
    objekA.sukaMembaca("Komik One Piece"); //memanggil method ke induk yang otomatis
diturunkan tanpa deklarasi ulang di anak
}    }

class Anak extends Ortu {
  void sukaMenonton(int a, String b) {
      System.out.println("Nonton Jam " + a + " Malam " + b);
  }
  void sukaMenonton(String a) {             // method induk spesifik
      System.out.println("Nonton " + a);
  }
  void sukaMembaca(String a) {     // method induk umum bisa diubah anak
     System.out.println("Suka Baca " + a);
  }
public static void main(String [] args) {
    System.out.println("Sifat Orang Tua :");
    Ortu objekO = new Ortu();     // memanggil objek induk
    objekO.sukaMenonton("Berita");        // memanggil sifat spesifik induk
    objekO.sukaMembaca("Koran");     // memanggil method dengan variabel dapat
diubah

    System.out.println("\n Sifat Anak :");
    Anak objekA = new Anak();    //memanggil objek anak
    objekA.sukaMenonton(9, "Film Drakor");        //memanggil sifat spesifik anak
yang diturunkan induk
    objekA.sukaMembaca("Komik One Piece"); //memanggil method ke induk yang otomatis
diturunkan tanpa deklarasi ulang di anak
}   }
```

Luaran 4:

```
Sifat Orang Tua :
Nonton Berita
Suka Baca Koran

 Sifat Anak :
Nonton Jam 9 Malam Film Drakor
Suka Baca Komik One Piece
```

Pada kode pemroraman di atas, Penulisan method belum efisien karena ada duplikasi kode yang tidak perlu. Pada class Anak, method sukaMenonton(String a) dideklarasikan kembali padahal sudah ada di class Ortu. Tidak perlu mendeklarasikan ulang method tersebut di class Anak jika tidak ada perubahan perilaku. Pada method sukaMembaca: Method ini juga dideklarasikan ulang di class Anak, padahal perilakunya sama dengan method di class Ortu.

[No.4] Analisis dan Argumentasi

Latihan 4:

4.1. Evaluasi method yang dimiliki `class Anak extends Ortu` dengan method di `class Ortu`!
Apakah penulisan method ini sudah efisien?

4.2. Setelah dirunning di JDoodle, catat waktu eksekusinya.
Rekomendasikan perbaikan penulisan kode method untuk dapat mengefisienkan waktu eksekusi!

Perbaikan yang dapat dilakukan pada kode program tersebut adalah menghapus method yang tidak perlu. Hapus sukaMenonton(String a) dan sukaMembaca(String a) di kelas Anak jika tidak ada perubahan dalam logika. Dengan cara ini, kita akan menghindari duplikasi dan mempermudah pemeliharaan kode. Kemudian, gunakan method dari kelas induk. Kelas Anak dapat menggunakan method dari Ortu tanpa mengoverride-nya, yang sudah diturunkan secara otomatis. Cukup panggil method dari Ortu di kelas Anak saat diperlukan. Waktu yang diperlukan untuk eksekusi sebelum perbaikan adalah 0,93 detik sedangkan setelah perbaikan hanya mebutuhkan 0,56 detik yang menunjukkan terjadinya efisiensi waktu dalam eksekusi.

Setelah perbaikan, **p**emanggilan method sukaMenonton(String a)dapat langsung dipanggil dari kelas Ortu, sehingga tidak perlu ada method yang didefinisikan ulang di kelas Anak. Pemanggilan method sukaMembaca(String a)dapat dipanggil langsung tanpa perlu mendeklarasikannya lagi di kelas Anak. Dengan method overloading, kita bisa menggunakan nama yang sama untuk berbagai operasi, tapi tetap memungkinkan metode untuk berfungsi dengan parameter yang berbeda. Dengan ini kode menjadi lebih ringkas dan mudah dibaca. Penggunaan pewarisan menghindari duplikasi dan mempermudah pemeliharaan di masa depan.

[No.4] Penyusunan Algoritma dan Kode Program

Rancang desain solusi atau algoritma

- Mulai program eclipse
- Definisikan sebuah class bernama Ortu
- a. Method sukaMenonton(String a)
- b. Method sukaMembaca(String a)
- Deklarasikan kelas Anak yang mewarisi kelas Ortu
- Method sukaMenonton(int a, String b)
- Di dalam method utama (main) kelas Ortu:
- a. Cetak "Sifat Orang Tua"
- b. Buat objek dari kelas Ortu
- c. Panggil method sukaMenonton("Berita") untuk objek Ortu
- d. Panggil method sukaMembaca("Koran") untuk objek Ortu
- e. Cetak "Sifat Anak"
- f. Buat objek dari kelas Anak
- g. Panggil method sukaMenonton(9, "Film Drakor") untuk objek Anak
- h. Panggil method sukaMembaca("Komik One Piece") yang diwarisi dari kelas Ortu
- Jalankan Program

Tuliskan kode program dan luaran

```java
package Rizqi.Nadhifah;

public class Ortu {
    void sukaMenonton(String a) {
        System.out.println("Nonton " + a);
    }
    void sukaMembaca(String a) {
        System.out.println("Suka Baca " + a);
    }
    public static void main(String[] args) {
        System.out.println("Sifat Orang Tua :");
        Ortu objekO = new Ortu();
        objekO.sukaMenonton("Berita");
        objekO.sukaMembaca("Koran");
        System.out.println("\nSifat Anak :");
        Anak objekA = new Anak();
        objekA.sukaMenonton(9, "Film Drakor");
        objekA.sukaMembaca("Komik One Piece");
    }}
class Anak extends Ortu {
    void sukaMenonton(int a, String b) {
        System.out.println("Nonton Jam " + a + " Malam " + b);
    }}
```

Problems Javadoc Declaration Console

<terminated> Ortu [Java Application] C:\Program Files\Java\jre1.8.0_421\bin\javaw.exe (Sep 2

```
Sifat Orang Tua :
Nonton Berita
Suka Baca Koran

Sifat Anak :
Nonton Jam 9 Malam Film Drakor
Suka Baca Komik One Piece
```

Luaran yang dihasilkan telah sesuai.

[No.4] Kesimpulan

Pada praktikum ini, Perbaikan yang dilakukan berfokus pada efisiensi kode dan penggunaan prinsip inheritance (pewarisan) dengan lebih optimal dalam konsep OOP (Object-Oriented Programming). Kode yang dihasilkan lebih bersih, tidak mengandung duplikasi method yang tidak perlu, dan memanfaatkan pewarisan dari kelas induk. Dengan demikian kode lebih singkat dan mudah dipelihara karena method yang tidak perlu dihapus. Kelas Anak mewarisi method dari Ortu tanpa perlu mendeklarasikan ulang method yang memiliki perilaku yang sama. Method overloading (perbedaan parameter pada method) masih dipertahankan untuk memberikan fungsionalitas tambahan pada kelas Anak. Dengan mengoptimalkan pewarisan dan menghindari duplikasi kode, program menjadi lebih mudah untuk dikembangkan dan dikelola.