

Nama & NPM	Topik:	Tanggal:
Nabila azizah Mutiara s G1F024031	Kelas, Objek, Method	17 september 2024

[No. 1] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variabel

```
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");
    }
}
```

Diketahui dari soal

- 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!

[No.1] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara memperbaiki nama konstruktor agar sesuai dengan nama kelas, menyimpan nilai parameter ke dalam atribut kelas, kemudian saya akan memastikan pembuatan objek menggunakan nama kelas yang benar.
- 2) Alasan solusi ini karena nama konstruktor harus sama dengan nama kelas untuk membangun objek dengan benar, atribut kelas harus diinisialisasi dengan nilai yang diterima oleh konstruktor agar data dapat disimpan dengan benar dan digunakan dalam objek dan juga nama kelas yang digunakan dalam pembuatan objek harus sesuai dengan nama kelas yang dideklarasikan.
- 3) Perbaiki kode program dengan cara ubah nama constructor dari Manusia1 menjadi Manusia agar sesuai dengan nama kelas, modifikasi constructor untuk menerima dua parameter, gunakan "this.nama" dan "this.ambut" dalam constructor untuk menginisialisasi atribut kelas dengan nilai yang diberikan oleh parameter. Ini memastikan bahwa atribut kelas memiliki nilai yang benar dan siap digunakan.

[No.1] Penyusunan Algoritma dan Kode Program

Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.

- a) Mulai
- b) Deklarasi Kelas
- c) Deklarasi atribut
- d) Deklarasi Constructor
- e) Metode main
- f) Selesai.

- a) Kode program dan luaran
- a) Screenshot/ Capture potongan kode dan hasil luaran
 Beri komentar pada kode yang di Screenshot
 1.1 Perbaiki pesan kesalahan Contoh 1!

The screenshot shows an Online Java Compiler IDE interface. On the left, the code editor displays a Java program with the following code:

```

1 public class Manusia { // deklarasi kelas
2     //deklarasi atribut Manusia dalam variabel
3     String nama, rambut;
4
5     //deklarasi constructor
6     public Manusia (String nama, String rambut) {
7         System.out.println(" Nama saya : " + nama +
8         "\n Warna Rambut : " + rambut);
9     }
10
11     //deklarasi method utama
12     public static void main( String[] args) {
13         Manusia satu = new Manusia("Putri", "hitam");
14     }
15 }
  
```

On the right, the 'Input/Output' panel shows the output of the program:

```

Nama saya : Putri
Warna Rambut : hitam
  
```

Below the output, the panel displays performance metrics: CPU Time: 0.05 sec(s) | Memory: 38896 kilobyte(s) | Compiled and executed in 1.243 sec(s).

- b) Analisa luaran yang dihasilkan
- 1.2 .

- a. **Atribut Variabel:** Atribut variabel (fields) adalah data yang dimiliki oleh objek dari kelas tersebut. Untuk kelas Manusia, atribut yang dapat diidentifikasi adalah:
- String nama: menyimpan nama orang.
 - String rambut: menyimpan warna rambut orang.
- b. **Perilaku/Behavior:** Perilaku (methods) adalah fungsi atau aksi yang dapat dilakukan oleh objek dari kelas tersebut. Dalam contoh ini:
- Konstruktor Manusia(String nama, String rambut): digunakan untuk menginisialisasi atribut nama dan rambut saat objek dibuat.
 - Method main(String[] args): berfungsi sebagai titik masuk program dan menampilkan informasi tentang objek Manusia

Secara keseluruhan, kode dan luaran sudah benar dan memenuhi tujuan yang diinginkan

[No.1] Kesimpulan

Analisa

- a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
- b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?
- jawaban Analisa:
- a) Program telah diperbaiki dengan benar sesuai dengan kebutuhan. Semua kesalahan pada kode awal telah diperbaiki dengan memastikan nama constructor sesuai dengan nama kelas dan parameter constructor sesuai dengan atribut yang perlu diinisialisasi. Kode yang diperbaiki dapat mengeluarkan output tanpa kesalahan
- b) Dengan perbaikan ini, program tidak hanya mengikuti aturan sintaksis Java tetapi juga memastikan fungsionalitasnya berfungsi dengan baik dan memberikan output yang benar sesuai dengan data yang dimasukkan.

[No. 2] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variabel

```
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");  
    }  
}
```

Diketahui dari soal

- 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!

[No.2] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara mengubah nama constructor menjadi nama kelas yang sama dan mendeklarasikan variabel nama dan rambut sebagai atribut kelas.
- 2) Alasan solusi ini karena adanya kesesuaian sintaks dan ases variable
- 3) Perbaiki kode program dengan cara mengubah Nama Constructor mengubah nama constructor dari public ortu menjadi public Ortu agar sesuai dengan nama kelas kemudian mendeklarasikan atribut, dan yang terakhir menginisialisasi Atribut dalam Constructor.

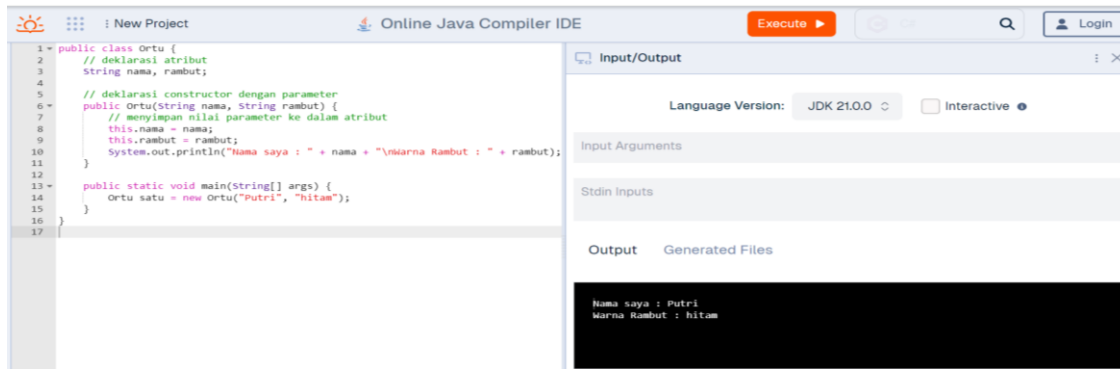
[No.1] Penyusunan Algoritma dan Kode Program

Algoritma

- a) Mulai
- b) Deklarasi Kelas dan Atribut
- c) Definisi Constructor
- d) Metode main
- e) Selesai.

- a) Kode program dan luaran

2.1 Evaluasi Penyebab Kesalahan dan Perbaiki Kode



2.2 Menambahkan Data Variabel Umur dan Jenis Kelamin



a) Screenshot/ Capture potongan kode dan hasil luaran
Beri komentar pada kode yang di Screenshot

b) Analisa luaran yang dihasilkan
pada gambar 2.1

Masalah:

1. Nama Constructor Salah: Nama constructor harus sama dengan nama kelas. Dalam kode yang Anda berikan, nama constructor adalah ortu, tetapi nama kelas adalah Ortu. Constructor harus memiliki nama yang sama dengan kelasnya.
2. Parameter Constructor Tidak Dideklarasikan: Constructor yang Anda deklarasikan tidak memiliki parameter, tetapi di main Anda mencoba membuat objek dengan parameter. Anda perlu mendefinisikan constructor dengan parameter yang sesuai.

Perbaiki:

1. Ubah nama constructor agar sesuai dengan nama kelas.
2. Tambahkan parameter pada constructor dan simpan nilai parameter tersebut dalam atribut kelas.

Pada gambar 2.2

Untuk menambahkan variabel umur dan jenisKelamin, yang saya lakukan adalah:

1. Menambahkan atribut baru pada kelas Ortu.
2. Menambahkan parameter untuk constructor agar dapat menerima nilai-nilai baru ini.
3. Menyimpan nilai parameter dalam atribut.

[No.2] Kesimpulan

(PILIH SALAH SATU ANDA INGIN MEMBAHAS DENGAN CARA ANALISA/ EVALUASI / KREASI)

1) Analisa

- a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
- b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

jawaban Analisa:

Pada program itu saya menggunakan bentuk kelas public karena kelas tersebut perlu diakses dari metode main yang bersifat static dalam lingkungan Java. Kelas public memungkinkan aksesibilitas yang lebih luas dari berbagai bagian program.

mengharuskan setiap konstruktor yang digunakan dalam instansiasi objek harus dideklarasikan dengan benar. Konstruktor harus didefinisikan untuk mencocokkan parameter yang diberikan saat pembuatan objek. Selain itu, variabel nama dan rambut dideklarasikan dalam kelas Ortu untuk menyimpan nilai yang diteruskan oleh konstruktor dan digunakan dalam output.

Dengan menambahkan konstruktor yang sesuai, program kini dapat menangani berbagai parameter dan menyimpan data tambahan seperti umur dan jenisKelamin, sesuai dengan kebutuhan yang ditentukan. Perbaikan ini memastikan bahwa program sesuai dengan struktur Java, dan dapat berjalan tanpa kesalahan kompilasi

[No. 3] Identifikasi Masalah:

1) permasalahan dan variabel

```
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");  
    }  
}
```

```
}  
}
```

Diketahui dari soal

- 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

[No.3] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara memperbaiki nama konstruktor agar sesuai dengan nama kelas, mengoreksi sintaks method yang tidak valid, dan menggunakan overload method dengan parameter berbeda.
- 2) Alasan solusi ini karena konstruktor harus memiliki nama yang sama dengan kelas, sintaks method harus benar agar dapat di-compile dan dieksekusi, dan metode overload memungkinkan fleksibilitas.
- 3) Perbaiki kode program dengan cara mengubah nama konstruktor dari Manusia1 menjadi Manusia untuk sesuai dengan nama kelas Manusia, Mengoreksi deklarasi method dengan menambahkan tanda kurung () dan menentukan tipe kembalian jika diperlukan. Metode yang tidak mengembalikan nilai harus dideklarasikan dengan tipe void, dan metode yang mengembalikan nilai harus memiliki tipe kembalian yang sesuai dengan nilai yang dikembalikan.

[No.3] Penyusunan Algoritma dan Kode Program

- a) Algoritma
 - (a) Mulai
 - (b) Identifikasi masalah
 - (c) Definisikan tujuan
 - (d) Rancang perubahan kode
 - (e) Implementasikan Perubahan
 - (f) Verifikasi Kode
 - (g) selesai
- a) Kode program dan luaran
 - 3.1. Evaluasi penyebab kesalahan dan perbaiki kode tersebut!

```

1 public class Manusia {
2     //deklarasi atribut Manusia dalam variabel
3     String nama, rambut;
4
5     //deklarasi constructor
6     public Manusia(String nama, String rambut) {
7         System.out.println(" Nama saya : " + nama +
8             "\n Warna Rambut : " + rambut);
9     }
10
11     //deklarasi method
12     void sukaNonton(String film){
13         System.out.println(" Hobi Menonton : " + film);
14     }
15
16     int sukaNonton (int episode, int durasi) {
17         return episode*durasi;
18     }
19
20     //deklarasi method utama
21     public static void main( String[] args) {
22         Manusia satu = new Manusia("Putri", "hitam");
23         satu.sukaNonton("Drakor");
24         int jumlahJam = satu.sukaNonton(2, 2);
25         System.out.println("Jam nonton = " + jumlahJam + " jam");
26     }
27 }

```

Input/Output

Language Version: JDK 21.0.0 ☐ Interactive

Input Arguments

Stdin Inputs

Output Generated Files

```

Nama saya : Putri
Warna Rambut : hitam
Hobi Menonton : Drakor
Jam nonton = 4 jam

```

CPU Time: 0.08 sec(s) | Memory: 40928 kilobyte(s) | Compiled and executed in 1.46 sec(s)

3.2. Ubahlah method dan constructor Contoh 3 sesuai dengan perilaku/ behavior anda

```

6 // Konstruktor dengan parameter
7 public Manusia(String nama, String rambut) {
8     this.nama = nama;
9     this.rambut = rambut;
10 }
11
12 // Method untuk memperkenalkan diri
13 void perkenalan() {
14     System.out.println("Nama saya: " + nama + "\nWarna Rambut: " + rambut);
15 }
16
17 // Method untuk menggambarkan hobi memasak kue
18 void sukaMemasakKue(String jenisKue) {
19     System.out.println("Hobi saya adalah memasak kue, terutama: " + jenisKue);
20 }
21
22 // Method untuk menghitung waktu memasak kue
23 int waktuMemasak(int jumlahKue, int waktuPerKue) {
24     return jumlahKue * waktuPerKue;
25 }
26
27 // Method utama
28 public static void main(String[] args) {
29     Manusia satu = new Manusia("Nabila", "hitam");
30
31     // Memperkenalkan diri
32     satu.perkenalan();
33
34     satu.sukaMemasakKue("Brownies");
35     int waktuJam = satu.waktuMemasak(5, 1); // Misalnya 5 kue, masing-masing
36     System.out.println("Waktu memasak = " + waktuJam + " jam");

```

Online Java Compiler IDE

Execute Bash Login

Language Version: JDK 21.0.0 ☒ Interactive

Input Arguments

Output Generated Files

```

Nama saya: Nabila
Warna Rambut: hitam
Hobi saya adalah memasak kue, terutama: Brownies
Waktu memasak = 5 jam

```

Compiled and executed in 1.985 sec(s)

c) Analisa luaran yang dihasilkan dan berikan komentar di setiap gambar pada gambar 3.1

Ada beberapa hal yang saya perbaiki mengenai hal hal yang menimbulkan kesalahan:

1. Nama Constructor Salah: Nama constructor dalam kelas Manusia harus sama dengan nama kelas itu sendiri. Anda menggunakan (Manusia1) sebagai nama constructor, sedangkan nama kelasnya adalah Manusia.
2. Sintaks Method Salah: Definisi method sukaNonton tidak benar. Method dalam Java harus dideklarasikan dengan tanda kurung dan tipe kembalian jika diperlukan.
3. Penggunaan Variabel yang Tidak Dideklarasikan variabel film tidak dideklarasikan dalam kelas.
4. Masalah Parameter Method: Method sukaNonton di dalam main mencoba untuk memanggil method dengan parameter, tetapi method tersebut tidak dideklarasikan.

Pada gambar 3.2

Penjelasan Perubahan:

1. **Konstruktor Manusia:** Konstruktor sekarang hanya menginisialisasi atribut nama dan rambut. Konstruktor ini diinisialisasi dengan nama "Nabila" dan rambut "hitam".
2. **Method perkenalan:** Method ini mencetak nama dan warna rambut.
3. **Method sukaMemasakKue:** Method ini mencetak informasi tentang hobi memasak kue dan jenis kue yang disukai.

4. **Method waktuMemasak:** Method ini menghitung waktu total yang diperlukan untuk memasak beberapa kue dengan waktu per kue yang diberikan.
5. **Method main:** Membuat objek Manusia dengan nama "Nabila" dan rambut "hitam", memanggil metode pengenalan untuk memperkenalkan diri, menggunakan metode sukaMemasakKue untuk menyebutkan hobi memasak kue, menghitung dan mencetak waktu yang diperlukan untuk memasak beberapa kue.

Penjelasan 3.3

Berikut merupakan hal hal yang dapat saya simpulkan

a) Constructor Overloading vs Overriding

- **Overloading:** adalah kemampuan untuk memiliki lebih dari satu constructor dalam satu kelas dengan parameter yang berbeda. Contoh: public Manusia() dan public Manusia(String nama, String rambut).
- **Overriding:** adalah konsep dalam pewarisan di mana method dalam subclass menggantikan method dalam superclass.

b) Method Overloading vs Overriding

- **Overloading:** Adalah kemampuan untuk memiliki beberapa method dengan nama yang sama tetapi parameter yang berbeda dalam satu kelas. Contoh: sukaNonton(String film) dan sukaNonton(int episode, int durasi).
- **Overriding:** Adalah kemampuan subclass untuk memberikan implementasi spesifik untuk method yang sudah ada di superclass dengan nama yang sama dan parameter yang sama.

c) Method yang Mengembalikan Nilai vs Tidak Mengembalikan Nilai

- **Method yang Mengembalikan Nilai:** Method yang memiliki tipe kembalian (seperti int, String, dll.) dan menggunakan return untuk mengembalikan nilai. Contoh: int sukaNonton(int episode, int durasi).
- **Method Tidak Mengembalikan Nilai:** Method yang memiliki tipe kembalian void dan tidak mengembalikan nilai apapun. Contoh: void sukaNonton(String film).

[No.3] Kesimpulan

1) Analisa

- a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
- b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

jawaban Analisa:

Perbaikan program dilakukan dengan mengganti Nama Konstruktor mengubah nama konstruktor Manusia1 menjadi Manusia untuk mencocokkan nama kelas, yang merupakan persyaratan sintaksis Java, mengoreksi Sintaks Metode menambahkan tanda kurung () pada deklarasi metode dan memastikan tipe kembalian yang sesuai. Hal ini diperlukan karena struktur Java mengharuskan semua metode untuk memiliki sintaks yang benar dan deklarasi yang tepat agar dapat dipanggil dan berfungsi dengan baik menambahkan Metode Baru: Menambahkan metode baru sukaMemasakKue dan waktuMemasak untuk mencerminkan perilaku yang diinginkan, yaitu hobi memasak kue. Perubahan ini memastikan bahwa kode memenuhi spesifikasi baru dan memberikan informasi yang relevan tentang hobi.

[No. 4] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variable


```

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 objek0 = new Ortu();        // memanggil objek induk
    objek0.sukaMenonton("Berita");    // memanggil sifat spesifik induk
    objek0.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 objek0 = new Ortu();          // memanggil objek induk
    objek0.sukaMenonton("Berita");      // memanggil sifat spesifik induk
    objek0.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
} }

```

Diketahui dari soal

- 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!

[No.4] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara menghapus metode yang tidak perlu di-override atau yang implementasinya identik dengan metode di kelas induk untuk mengurangi redundansi dan meningkatkan efisiensi kode

- 2) Alasan solusi ini karena mengurangi redundansi kode membuat kode lebih bersih dan mudah dikelola. Ini juga meningkatkan efisiensi dengan menghindari duplikasi implementasi yang tidak perlu, yang pada gilirannya dapat mempengaruhi waktu eksekusi dan pemeliharaan kode.
- 3) Perbaiki kode program dengan cara menghapus metode yang tidak perlu di kelas anak jika implementasinya sama dengan kelas induk, serta memastikan hanya metode yang benar-benar diubah atau ditambahkan yang tetap ada. Ini memastikan kode lebih efisien, lebih mudah dibaca, dan lebih mudah dipelihara.

[No.4] Penyusunan Algoritma dan Kode Program

a) Algoritma

- (a) Mulai
- (b) Inisialisasi
- (c) Identifikasi dan Analisis
- (d) Desain Perbaikan
- (e) Implementasi Perbaikan
- (f) Verifikasi
- (g) selesai

b) Kode program dan luaran

4.2

The screenshot shows an Online Java Compiler IDE with the following code in the editor:

```

1 public class Ortu { // membuat kelas induk
2     void sukaMenonton(String a) { // method induk spesifik
3         System.out.println("Nonton " + a);
4     }
5     void sukaMembaca(String a) { // method induk umum bisa diubah anak
6         System.out.println("Suka Baca " + a);
7     }
8
9     public static void main(String [] args) {
10        System.out.println("Sifat Orang Tua :");
11        Ortu objekO = new Ortu();
12        objekO.sukaMenonton("Berita");
13        objekO.sukaMembaca("Koran");
14
15        System.out.println("\nSifat Anak :");
16        Anak objekA = new Anak(); // memanggil objek anak
17        objekA.sukaMenonton(9, "Film Drakor"); // memanggil sifat spesifik anak
18        objekA.sukaMembaca("Komik One Piece"); // memanggil method ke induk yang otomatis
19    }
20 }
21
22 class Anak extends Ortu {
23     void sukaMenonton(int a, String b) {
24         System.out.println("Nonton Jam " + a + " Malam Film " + b);
25     }
26 }
27

```

The Output window shows the following results:

```

Sifat Orang Tua :
Nonton Berita
Suka Baca Koran

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

```

At the bottom, it states: "Compiled and executed in 1.844 sec(s)"

Analisa luaran yang dihasilkan dan berikan komentar

4.1 Analisis Method pada Kelas Anak dan Kelas Ortu:

- Kelas Ortu:
 - sukaMenonton(String a): Mencetak pesan dengan satu parameter string.
 - sukaMembaca(String a): Mencetak pesan dengan satu parameter string.
- Kelas Anak:
 - sukaMenonton(int a, String b): Mencetak pesan dengan dua parameter, satu integer dan satu string.
 - sukaMenonton(String a): Overriding dari method yang sama di kelas Ortu.
 - sukaMembaca(String a): Mengoverride method dari kelas Ortu, meskipun implementasinya identik dengan kelas Ortu.
- Kurang Efisien:

Kelas Anak mengoverride `sukaMembaca` tanpa perubahan pada implementasi, yang menambah kompleksitas tanpa keuntungan tambahan.

Implementasi `sukaMenonton(String a)` di kelas Anak yang identik dengan implementasi di kelas Ortu menambah redundansi.

4.2 Rekomendasi untuk Efisiensi:

1. Hapus Overriding yang Tidak Perlu: Jika implementasi method di kelas Anak tidak berbeda dari implementasi di kelas Ortu, maka tidak perlu menuliskannya ulang di kelas Anak.
2. Penghapusan Metode yang Redundan: Hapus method `sukaMembaca(String a)` di kelas Anak jika implementasinya tidak berubah.
3. Optimasi Code: Menjaga kode lebih bersih dan lebih mudah dikelola dengan menghindari pengulangan kode yang tidak perlu
Waktu mengerjakan tercantum di dalam gambar.

1) Analisa

- a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
- b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

jawaban Analisa

Pada program itu saya menggunakan bentuk kelas public karena kelas public diperlukan untuk memungkinkan akses ke kelas dari luar paket. Dalam hal ini, baik kelas Ortu maupun Anak harus diakses dari metode main, sehingga keduanya harus dideklarasikan sebagai public. Perbaikan program dengan menambahkan perbaikan dilakukan dengan menghapus metode yang identik di kelas Anak yang sudah ada di kelas Ortu. Ini karena struktur Java mengharuskan kode yang bersih dan efisien tanpa duplikasi yang tidak perlu. Menambahkan kode yang redundant tidak hanya membuat

Refleksi

Pada minggu ini, pengalaman belajar saya berfokus pada pemahaman mendalam mengenai konsep-konsep dasar dalam pemrograman Java, khususnya terkait dengan objek, metode, dan kelas melalui penggunaan metode overriding dan overloading. Kegiatan ini memberikan wawasan yang berharga mengenai bagaimana struktur kode dapat mempengaruhi efisiensi dan keterbacaan program.