Jawaban:1

Nama & NPM	Topik:	Tanggal:
Fikri irwansyah G1F024073	**Unit 1(class)**	**18 september 2024**

[No. 1] Identifikasi Masalah:

1) Uraikan permasalahan dan variabel Uraikan permasalahan dan variable.

   Mengelola variabel dengan baik melibatkan pemahaman tentang cara kerjanya dalam lingkup (scope), tipe data, dan perilaku variabel dalam Java. Dengan memperhatikan penamaan yang tepat, inisialisasi yang benar, dan pemilihan scope yang sesuai, permasalahan terkait variabel dapat dihindari.

Tuliskan kembali soal:

Analisa ciri-ciri umum Kelas Manusia yang dapat menjadi

a. atribut variabel, dan

b. perilaku/ behavior untuk method!

Pada soal masih ada pesan kesalahan

Kemungkinan kesalahan yang sering terjadi dalam kode Java ini mungkin terkait dengan:

1. Penggunaan variabel yang tidak diinisialisasi dengan benar.
2. Hak akses variabel yang belum diatur.
3. Perbedaan antara penulisan nama file dan nama kelas (harus sama di Java).

Atau
Diketahui dari soal : variabel dari dalam kelas

[No.1] Analisis dan Argumentasi

1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara

   Permasalahan variabel dalam kelas `Manusia` dapat diselesaikan dengan

   beberapa cara:

   A.Enkapsulasi untuk menjaga keamanan akses variabel.

   B.Inisialisasi default untuk mencegah nilai tidak terdefinisi.

   C.Static variabel untuk data yang bersifat umum bagi semua objek.

   D.Validasi input untuk memastikan data yang valid..

2) Alasan solusi ini karena :

Permasalahan variabel dalam kelas `Manusia` dapat diselesaikan dengan beberapa cara:

A.Enkapsulasi untuk menjaga keamanan akses variabel.

B.Inisialisasi default untuk mencegah nilai tidak terdefinisi.

C.Static variabel untuk data yang bersifat umum bagi semua objek.

D.Validasi input untuk memastikan data yang valid.

E.Penggunaan helper class untuk menyederhanakan kode.

3) Perbaikan kode program dengan cara:

- Konstruksi dan Penggunaan Objek**:** Program utama membuat objek manusia baik dengan constructor tanpa parameter maupun dengan parameter. Ini menunjukkan bagaimana objek dapat digunakan dan dimodifikasi.
- Validasi Input**:** Contoh penggunaan setter dan constructor yang melemparkan pengecualian jika input tidak valid. Hal ini penting untuk memastikan bahwa data yang dimasukkan selalu valid dan konsisten.

[No.1] Penyusunan Algoritma dan Kode Program

1) Algoritma

- Inisialisasi Kelas**:** Menetapkan nilai default dan melakukan validasi pada konstruktor dan setter.
- Metode Getter dan Setter**:** Mengatur akses dan modifikasi variabel instance dengan validasi.
- Metode compli**():** Menampilkan informasi tentang objek.
- Program Utama**:** input yang tidak valid.

Menggunakan kelas `Manusia` untuk membuat objek, memodifikasi data, dan menangani

2) Kode program dan luaran

```
public class Manusia {  // deklarasi kelas
    // deklarasi variabel
    String nama;
    String rambut;
     // deklarasi constructor tanpa parameter
     public Manusia() {
          System.out.println("Kelas Manusia tanpa nama");
     }
  }
```

Refleksi

(saya harus masih banyak mempelajari materi ini karena masih sulit di pahami)

Nama & NPM	Topik:	Tanggal:
Fikri irwansyah G1F024073	UNIT 2:OBJEK	**18 september 2024**

[Nomor Soal] Identifikasi Masalah:

1) Uraikan permasalahan dan variable

Masalah-masalah umum yang terkait dengan variabel seringkali berkaitan dengan skop, tipe data, atau penulisan nama yang tidak benar. Untuk menghindari masalah ini, penting untuk mempraktikkan penulisan kode yang rapi, memilih nama variabel yang deskriptif, dan memahami aturan skop variabel di bahasa pemrograman yang digunakan.

2) Rincikan sumber informasi yang relevan (buku / webpage)
Web page:URL**:** https://docs.oracle.com/javase/tutorial/java/nutsandbolts/variables.html

[Nomor Soal] Analisis dan Argumentasi

1) Uraikan rancangan solusi yang diusulkan.
Rancangan yang saya usulkan ketika terjadi kesalahan menanyakan kepada teman yg lebih memahami ketika terjadi kesalahan atau menggunakan IDE atu linting tools
2) Analisis solusi, kaitkan dengan permasalahan.
Solusi nya saat terjadi kesalahan memahami ulang materinya

[Nomor Soal] Penyusunan Algoritma dan Kode Program

1) Rancang desain solusi atau algoritma

Algoritma**:**

1. Tentukan skop (ruang lingkup) di mana variabel akan digunakan.
2. Jika variabel hanya dibutuhkan dalam satu fungsi, deklarasikan variabel sebagai variabel lokal.
3. Jika variabel dibutuhkan di seluruh kelas, deklarasikan sebagai variabel instance atau global.
4. Gunakan this untuk membedakan variabel instance dari variabel lokal jika diperlukan.

2) Tuliskan kode program dan luaran

```
public class Ortu {
   //deklarasi constructor
   public Ortu(String nama, String rambut)  {
       //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("Fikri", "hitam");
    }
}
```

Luaran:
Nama saya : Fikri
Warna Rambut : hitam

=== Code Execution Successful ===

a) Beri komentar pada kode
Nama saya : Fikri
Warna Rambut : hitam

=== Code Execution Successful ===

b) Uraikan luaran yang dihasilkan

Luuaran:
Nama saya : Fikri
Warna Rambut : hitam

=== Code Execution Successful ===

c) Screenshot/ Capture potongan kode dan hasil luaran

```
public class Ortu {
    //deklarasi constructor
    public Ortu(String nama, String rambut)  {
        //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("Fikri", "hitam");
    }
}
```

```
java -cp /tmp/6zTJohosck/Ortu
 Nama saya : Fikri
 Warna Rambut : hitam

=== Code Execution Successful ===
```

[Nomor Soal] Kesimpulan

1) Kreasi

a) Apakah ada pengetahuan baru yang dikembangkan dan konsep baru sebagai usulan solusi?ada pengetahuan baru yang saya dapat walalupun sedikit tetapi saya akan berusaha untuk mempelajari lagi

b) Konstruksikan hubungan antara variabel yang berbeda dengan konsep yang anda ketahui! (jika ada)

Refleksi:saya dapat pengetahuan baru dikit demi sedikit

Jawaban:2

Nama & NPM	Topik:	Tanggal:
Fikri irwansyah G1F024073	UNIT 3:METHOD	**18 september 2024**

[Nomor Soal] Identifikasi Masalah:

1)Uraikan permasalahan dan v7ariable

Permasalahan utama dalam kode ini adalah bahwa konstruktor tidak sesuai dengan nama kelas, serta variabel instance nama dan rambut tidak disimpan dengan benar dari parameter konstruktor. Penggunaan kata kunci this dan penamaan yang tepat akan menyelesaikan masalah ini, sehingga variabel instance dapat digunakan dengan benar dalam seluruh kelas.

2)Rincikan sumber informasi yang relevan (buku / webpage)
JavaTPoint: Java Constructors

3)Uraikan rancangan solusi yang diusulkan (jika ada).

4)Analisis susunan solusi, parameter solusi (jika ada).

[Nomor Soal] Analisis dan Argumentasi

5)Uraikan rancangan solusi yang diusulkan.

A.Perbaiki nama konstruktor agar sesuai dengan kelas.

B. Gunakan `this` untuk memastikan parameter konstruktor disimpan ke dalam variabel instance

C. Inisialisasi variabel instance dengan nilai yang sesuai, dan pastikan skop variabel tepat.

[Nomor Soal] Penyusunan Algoritma dan Kode Program

7)Tuliskan kode program dan luaran

a)Beri komentar pada kode

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

   // deklarasi constructor dengan nama kelas yang sesuai
   public Manusia(String nama, String rambut) {
      // simpan nilai parameter ke atribut instance
      this.nama = nama;
      this.rambut = rambut;
      System.out.println("Nama saya : " + this.nama +
         "\nWarna Rambut : " + this.rambut);
   }

   // deklarasi method
   void hobiSaya(String hobi) {
      System.out.println("Hobi : " + hobi);
   }
```

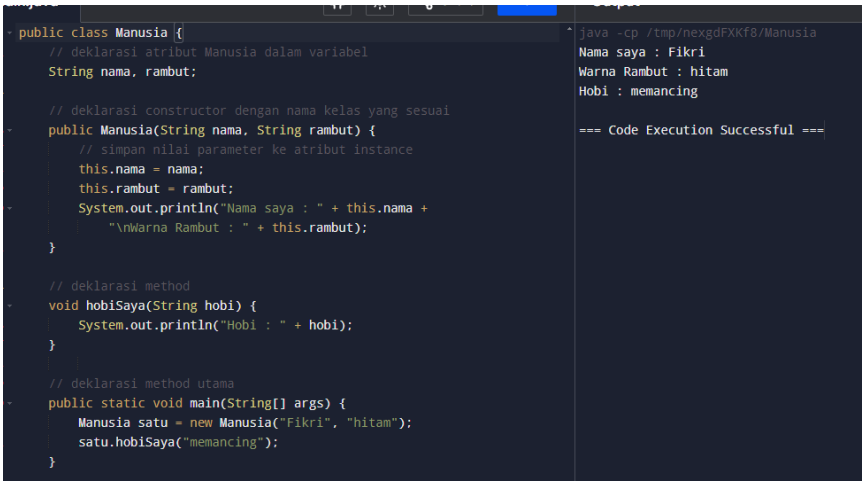
```
    // deklarasi method utama
    public static void main(String[] args) {
        Manusia satu = new Manusia("Fikri", "hitam");
        satu.hobiSaya("memancing");
    }
}
```

b)Uraikan luaran yang dihasilkan

Nama saya : Fikri
Warna Rambut : hitam
Hobi : memancing

=== Code Execution

C.screenshoot/Capture potongan kode dan hasil luaran



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

    // deklarasi constructor dengan nama kelas yang sesuai
    public Manusia(String nama, String rambut) {
        // simpan nilai parameter ke atribut instance
        this.nama = nama;
        this.rambut = rambut;
        System.out.println("Nama saya : " + this.nama +
            "\nWarna Rambut : " + this.rambut);
    }

    // deklarasi method
    void hobiSaya(String hobi) {
        System.out.println("Hobi : " + hobi);
    }

    // deklarasi method utama
    public static void main(String[] args) {
        Manusia satu = new Manusia("Fikri", "hitam");
        satu.hobiSaya("memancing");
    }
}
```

```
java -cp /tmp/nexgdFXKf8/Manusia
Nama saya : Fikri
Warna Rambut : hitam
Hobi : memancing

=== Code Execution Successful ===
```

Uraikan perbedaan berikut:

a) constructor overloading dan overriding

b) method overloading, dan method overriding

c) method yang mengembalikan nilai dan method tidak mengembalikan nilai

A. Konstruktor Overloading: Memungkinkan banyak konstruktor dalam satu kelas dengan parameter yang berbeda.
B.Method Overloading**:** Memungkinkan banyak metode dengan nama yang sama tetapi parameter yang berbeda dalam satu kelas.
C. Method Overriding**:** Menyediakan implementasi baru untuk metode yang diwarisi dari superclass.
D.Method yang Mengembalikan Nilai**:** Mengembalikan nilai dengan tipe data yang dideklarasikan.
E. Method yang Tidak Mengembalikan Nilai**:** Tidak mengembalikan menggunakan `void` sebagai tipe pengembalian.

[Nomor Soal] Kesimpulan

2) Evaluasi

a) Apa konsekuensi dari skenario pemprograman ini?

Konsukuensi memungkinkan sering terjadi kesalahan yang tidak di ketahui dimana letak posisinya

b) Evaluasi input, proses, dan luaran yang dihasilkan! (jika ada)

	Proses hasilnya success Refleksi :lebih banyak mendalami hal yang perlu di pahami agar lebih mudah untuk memahami apa yang di pelajari

Jawaban:3

Nama & NPM	Topik:	Tanggal:
Fikri irwansyah G1F024073	UNIT 4:EXTENDS	**18 september 2024**

[Nomor Soal] Identifikasi Masalah:

1)Uraikan permasalahan dan variable
Tidak ada permasalahan di variable ini
2)Rincikan sumber informasi yang relevan (buku / webpage)
Oracle Java Documentation

3) Uraikan rancangan solusi yang diusulkan (jika ada).
4) Analisis susunan solusi, parameter solusi (jika ada).
5) Bandingkan method yang dimiliki `class Anak extends Ortu` dengan method di `class Ortu!`

```
public class Ortu {
    void sukaMenonton(String a) {    // Method di kelas Ortu
        System.out.println("Nonton " + a);
    }

    void sukaMembaca(String a) {     // Method di kelas Ortu
        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 metode spesifik induk
   objekO.sukaMembaca("Koran");     // Memanggil metode umum induk

        System.out.println("\nSifat Anak :");
        Anak objekA = new Anak();    // Memanggil objek anak
        objekA.sukaMenonton(9, "Film Drakor");        // Memanggil metode overload
        objekA.sukaMenonton("Drama"); // Memanggil metode override
        objekA.sukaMembaca("Komik One Piece"); // Memanggil metode override
    }
}
```

```
class Anak extends Ortu {
```

```
    void sukaMenonton(int a, String b) {
      System.out.println("Nonton Jam " + a + " Malam " + b);
    }

    @Override
    void sukaMenonton(String a) {    // Override method dari Ortu
      System.out.println("Nonton " + a);
    }

    @Override
    void sukaMembaca(String a) {     // Override method dari Ortu
      System.out.println("Suka Baca " + a);
    }
  }
```

[Nomor Soal] Analisis dan Argumentasi

5)Uraikan rancangan solusi yang diusulkan.
6)Analisis solusi, kaitkan dengan permasalahan.

[Nomor Soal] Penyusunan Algoritma dan Kode Program

7)Rancang desain solusi atau algoritma
8)Tuliskan kode program dan luaran

```
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("tv");    // memanggil sifat spesifik induk
    objekO.sukaMembaca("majalah berita");     // memanggil method dengan variabel
dapat diubah

    System.out.println("\n Sifat Anak :");
    Anak objekA = new Anak();    //memanggil objek anak
    objekA.sukaMenonton(9, "Film kartun di youtube");       //memanggil sifat spesifik anak
yang diturunkan induk
    objekA.sukabermain("sepeda"); //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 sukabermain(String a) {     // method induk umum bisa diubah anak
```

```
        System.out.println("Suka bermain " + 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:

```
Sifat Orang Tua :
Nonton tv
Suka Baca majalah berita

 Sifat Anak :
Nonton Jam 9 Malam Film kartun di youtube
Suka bermain sepeda

=== Code Execution Successful ===
```

d) Beri komentar pada kode

```
Kode sucses
```

e) Uraikan luaran yang dihasilkan

```
Sifat Orang Tua :
Nonton tv
Suka Baca majalah berita

 Sifat Anak :
Nonton Jam 9 Malam Film kartun di youtube
Suka bermain sepeda

=== Code Execution Successful ===
```

f) Screenshot/ Capture potongan kode dan hasil luaran

```
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("tv");    // memanggil sifat spesifik induk
```

```
   objekO.sukaMembaca("majalah berita");     // memanggil method dengan variabel
dapat diubah

   System.out.println("\n Sifat Anak :");
   Anak objekA = new Anak();    //memanggil objek anak
   objekA.sukaMenonton(9, "Film kartun di youtube");       //memanggil sifat spesifik
anak yang diturunkan induk
   objekA.sukabermain("sepeda"); //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 sukabermain(String a) {     // method induk umum bisa diubah anak
      System.out.println("Suka bermain " + 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:
Sifat Orang Tua :
Nonton tv
Suka Baca majalah berita

 Sifat Anak :
Nonton Jam 9 Malam Film kartun di youtube
Suka bermain sepeda

=== Code Execution Successful ===
```

[Nomor Soal] Kesimpulan

1)Kreasi

a.Apakah ada pengetahuan baru yang dikembangkan dan konsep baru sebagai usulan solusi?
Saya mendapatkan pengetahuan sedikit demi sedikit dan saya masih banyak harus mempelajari lagi lebih dalam

B.Konstruksikan hubungan antara variabel yang berbeda dengan konsep yang anda ketahui! (jika ada) Refleksi:berusaha,belajar dan berkembang

Jawaban:4