|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Khusnul Hidayati**  **G1F024032** | **Kelas, Objek, Method** | **17 September 2024** |

**[No. 1] Identifikasi Masalah:**

1. Uraikan permasalahan dan variabel

Jawab:

**Contoh 1:**  
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)

**Latihan 1:**  
1.1.     Perbaiki pesan kesalahan Contoh 1!

**Jawab:**

**Mengubah** Manusia1 satu = new Manusia1("Putri", "hitam"); karena tidak sesuai dengan deklarasi kelas awal, jadi diubah menjadi manusia satu = **new** manusia("Putri", "hitam"); agar sesuai dengan deklarasi kelas awal dan program tidak error.  
1.2.     Analisa ciri-ciri lain Kelas Manusia yang dapat menjadi  
           a. atribut variabel, dan

Jawab:

Umur, Jenis Kelamin, Tinggi Badan, Warna Kulit, Hobi, dan lain – lain.

 b. perilaku/ behavior!

Jawab:

Belajar, Bersosialisasi, dan lain – lain.

**[No.1] Analisis dan Argumentasi**

Jawab:

1. Saya mengusulkan permasalahan ini dapat diatasi dengan cara **mengubah baris** Manusia1 satu = new Manusia1("Putri", "hitam"); karena tidak sesuai dengan deklarasi kelas awal, jadi diubah menjadi manusia satu = **new** manusia("Putri", "hitam");
2. Alasan solusi ini karena agar sesuai dengan deklarasi kelas awal dan program tidak error.

**[No.1 ] Penyusunan Algoritma dan Kode Program**

1. Algoritma

Jawab:

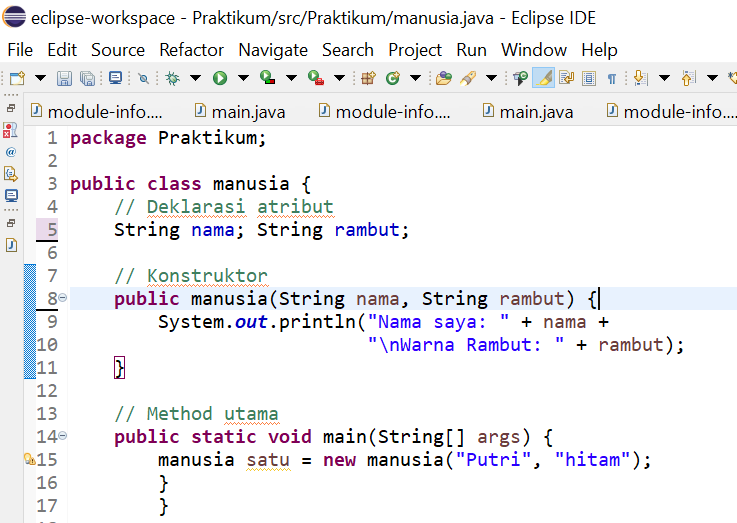
Algoritma adalah langkah-langkah penyelesaian masalah.

1. **Mengubah baris** Manusia1 satu = new Manusia1("Putri", "hitam"); karena tidak sesuai dengan deklarasi kelas awal, jadi diubah menjadi manusia satu = **new** manusia("Putri", "hitam");
2. Kode program dan luaran

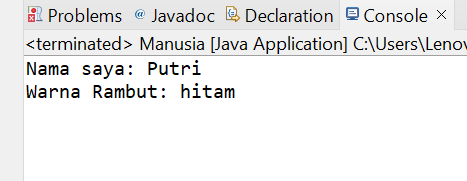
Jawab:

1. Screenshot/ Capture potongan kode dan hasil luaran

Beri komentar pada kode yang di Screenshot



1. Analisa luaran yang dihasilkan   
   Jawab:



Luaran telah sesuai dengan program yang disusun.   
Tipe data yang ditampilkan juga sudah sesuai dengan kebutuhan dan permintaan data.

**[No.1] Kesimpulan**

1. **Analisa**

Jawab:

1. Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
2. Permasalahan

* Pada baris Manusia1 satu = new Manusia1("Putri", "hitam"); belum sesuai dengan deklarasi kelas awal.

1. Algoritma

* **Mengubah baris** Manusia1 satu = new Manusia1("Putri", "hitam"); karena tidak sesuai dengan deklarasi kelas awal, jadi diubah menjadi manusia satu = **new** manusia("Putri","hitam");

1. Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

Jawab:

Dasar pengambilan keputusan ini adalah agar sesuai dengan deklarasi kelas awal dan program tidak error serta dapat berjalan.

**[No. 2] Identifikasi Masalah:**

1. Uraikan permasalahan dan variabel

Jawab:

**Contoh 2**: Salin dan tempel kode program berikut ke Eclipse atau JDoodle.

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)

**Latihan 2:**  
2.1.  Evaluasi penyebab kesalahan dan perbaiki kode tersebut!

Jawab:

Menambahkan (String nama, String rambut) pada baris     public ortu   {    
  
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!

Jawab:

"\nUmur: " + umur +

"\nJenis Kelamin: " + (jenisKelamin == 'P' ? "Perempuan" : "Laki-laki"));

**[No.2] Analisis dan Argumentasi**

1. Saya mengusulkan permasalahan ini dapat diatasi dengan cara menambahkan (String nama, String rambut) pada baris     public ortu   {   dan menambahkan baris untuk constructor dengan parameter baru.
2. Alasan solusi ini karena agar program tidak error dan bisa berjalan.

**[No.2 ] Penyusunan Algoritma dan Kode Program**

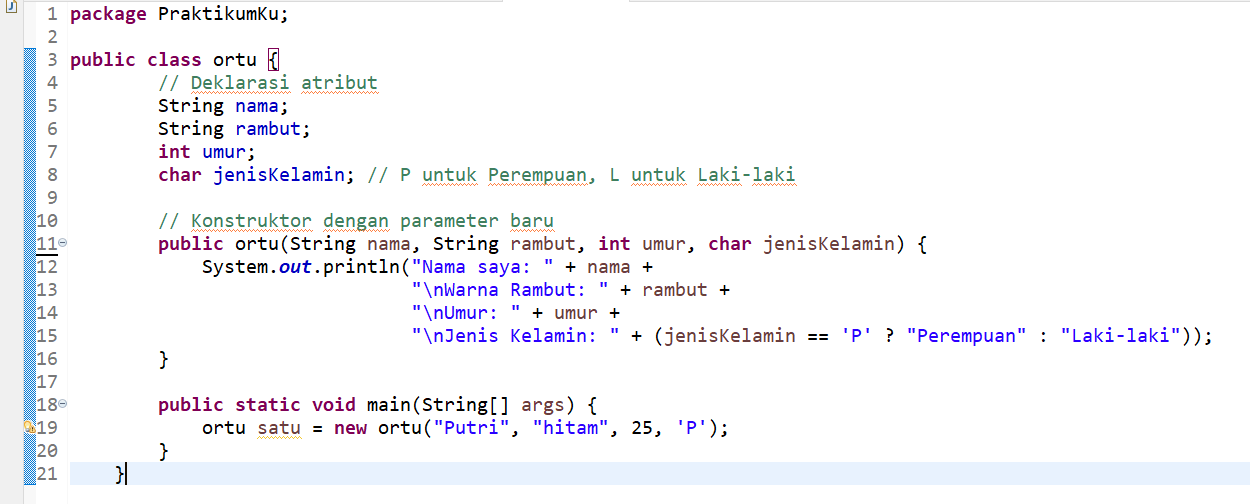
1. Algoritma

Jawab:

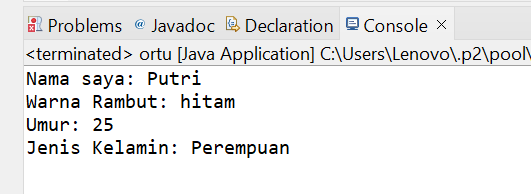
Algoritma adalah langkah-langkah penyelesaian masalah.

1. Menambahkan (String nama, String rambut) pada baris     public ortu   {
2. Menambahkan baris untuk constructor dengan parameter baru.
3. Kode program dan luaran
4. Screenshot/ Capture potongan kode dan hasil luaran

Beri komentar pada kode yang di Screenshot



1. Analisa luaran yang dihasilkan   
   Jawab:



Luaran telah sesuai dengan program yang disusun.   
Tipe data yang ditampilkan juga sudah sesuai dengan kebutuhan dan permintaan data.

**[No.2] Kesimpulan**

1. **Analisa**

Jawab:

1. Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
2. Permasalahan

* Pada baris     public ortu   {   belum ada (String nama, String rambut)

1. Algoritma

* Menambahkan (String nama, String rambut) pada baris     public ortu   {

1. Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

Jawab:

Dasar pengambilan keputusan ini adalah agar program dapat berjalan dan tidak error.

**[No. 3] Identifikasi Masalah:**

1. Uraikan permasalahan dan variabel

Jawab:

**Contoh 3:** Salin dan tempel kode program berikut ke Eclipse atau JDoodle.  
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)

**Latihan 3:**  
3.1.  Evaluasi penyebab kesalahan dan perbaiki kode tersebut!

Jawab:

* Menambahkan (String film) pada baris     void sukaNonton  {
* Mengubah int sukaNonton { menjadi **int** sukaNonton(**int** episode, **int** durasi) {

episode\*durasi; **return** episode\*durasi;

untuk menghitung durasi menonton.  
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

Jawab:

1. Constructor overloading terjadi ketika beberapa constructor berada dalam satu kelas dengan nama dan parameter yang berbeda. Sedangkan constructor overriding tidak ada, karena setiap kelas memiliki constructornya masing – masing.
2. Method overloading memungkinkan beberapa metode dengan nama yang sama berada di kelas yang sama, tetapi berbeda dalam jumlah atau tipe parameter. Sedangkan method overriding subclass memiliki kemampuan untuk menyediakan implementasi baru untuk metode yang sudah ada di superclass yang memiliki nama dan parameter yang sama.
3. Method yang mengembalikan nilai dideklarasikan dengan tipe kembalian dan menggunakan return untuk mengembalikan nilai. Sedangkan method tidak mengembalikan nilai dideklarasikan dengan void dan tidak mengembalikan nilai. Hanya menjalankan instruksi di dalamnya.

**[No.3] Analisis dan Argumentasi**

1. Saya mengusulkan permasalahan ini dapat diatasi dengan cara menambahkan (String film) pada baris     void sukaNonton  { dan mengubah

int sukaNonton { menjadi **int** sukaNonton(**int** episode, **int** durasi) { episode\*durasi; **return** episode\*durasi;

untuk menghitung durasi menonton.

1. Alasan solusi ini karena agar program dapat berjalan dan tidak error.

**[No.3 ] Penyusunan Algoritma dan Kode Program**

1. Algoritma

Jawab:

Algoritma adalah langkah-langkah penyelesaian masalah.

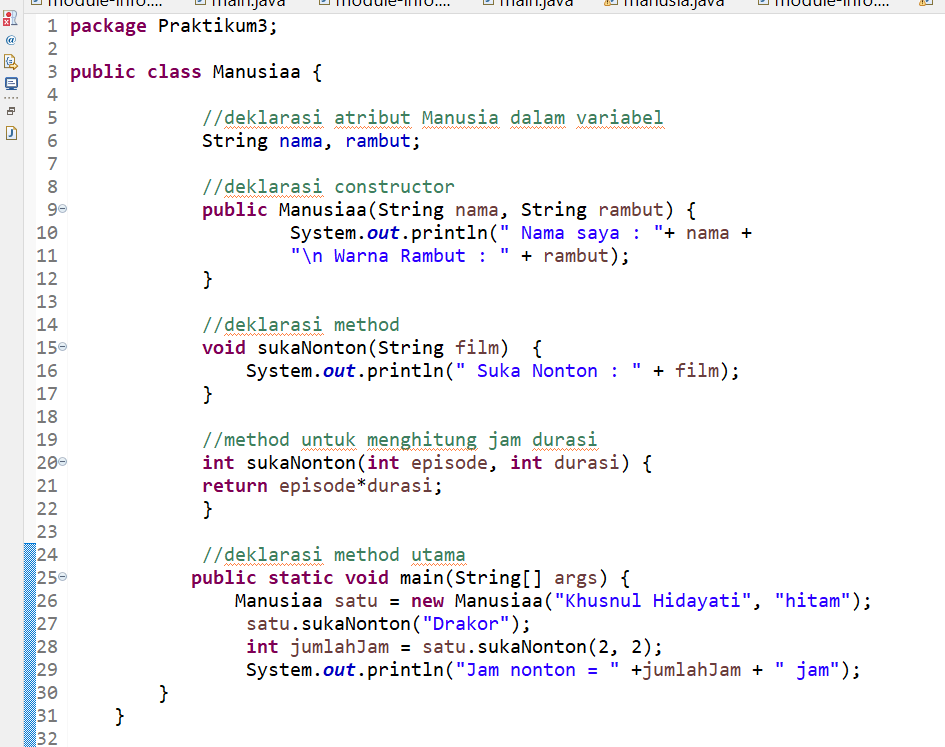
1. Menambahkan (String film) pada baris     void sukaNonton  {
2. Mengubah int sukaNonton { menjadi **int** sukaNonton(**int** episode, **int** durasi) {

episode\*durasi; **return** episode\*durasi;

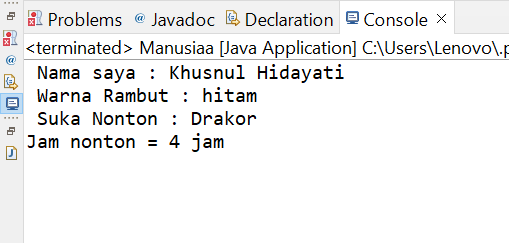
untuk menghitung durasi menonton.

1. Mengubah method dan constructor Contoh 3 sesuai dengan perilaku/ behavior saya.
2. Kode program dan luaran
3. Screenshot/ Capture potongan kode dan hasil luaran

Beri komentar pada kode yang di Screenshot



1. Analisa luaran yang dihasilkan   
   Jawab:



Luaran telah sesuai dengan program yang disusun.   
Tipe data yang ditampilkan juga sudah sesuai dengan kebutuhan dan permintaan data.

**[No.3] Kesimpulan**

1. **Analisa**

Jawab:

1. Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
2. Permasalahan

* Pada baris     void sukaNonton  { belum ada (String film)
* int sukaNonton {diubah menjadi **int** sukaNonton(**int** episode, **int** durasi) {

episode\*durasi; **return** episode\*durasi;

1. Algoritma

* Menambahkan (String film) pada baris     void sukaNonton  {
* Mengubah int sukaNonton { menjadi **int** sukaNonton(**int** episode, **int** durasi) {

episode\*durasi; **return** episode\*durasi;

untuk menghitung durasi menonton.

1. Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

Jawab:

Dasar pengambilan keputusan ini adalah agar program dapat dijalankan dan tidak error.

**[No. 4] Identifikasi Masalah:**

1. Uraikan permasalahan dan variabel

Jawab:

**Contoh 4:** Salin dan tempel kode program berikut ke JDoodle. Kemudian catat waktu eksekusinya.

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

**Latihan 4:**  
4.1. Evaluasi method yang dimiliki class Anak extends Ortu dengan method di class Ortu!  
       Apakah penulisan method ini sudah efisien?

Jawab:

Belum, karena banyak menggunakan method yang sama dengan perilaku yang tidak berbeda membuat kode lebih sulit dibaca.  
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 perbaikan penulisan kode method.
2. Alasan solusi ini karena untuk mengefisienkan waktu eksekusi.

**[No.4 ] Penyusunan Algoritma dan Kode Program**

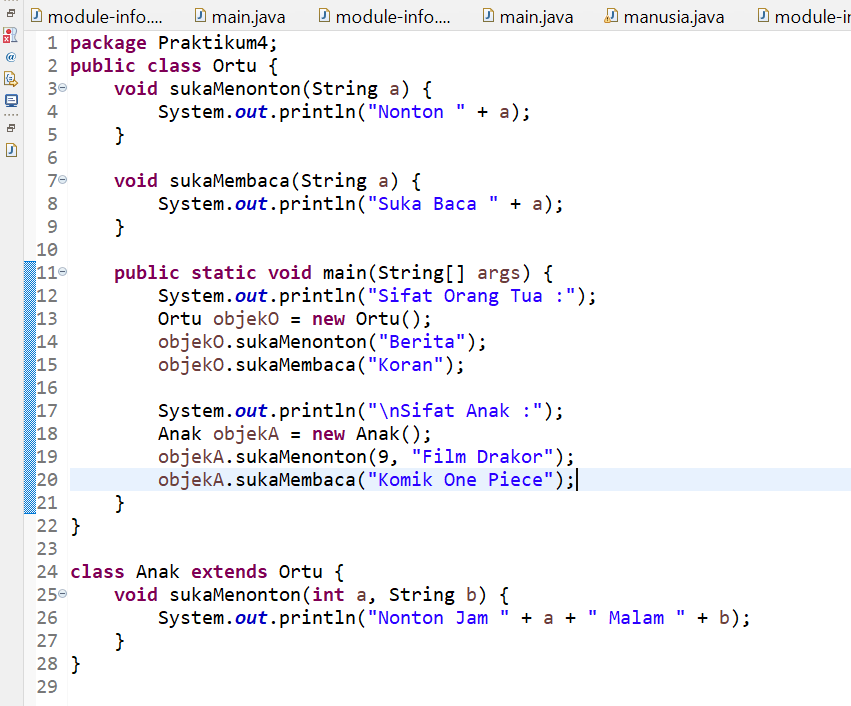
1. Algoritma

Jawab:

Algoritma adalah langkah-langkah penyelesaian masalah.

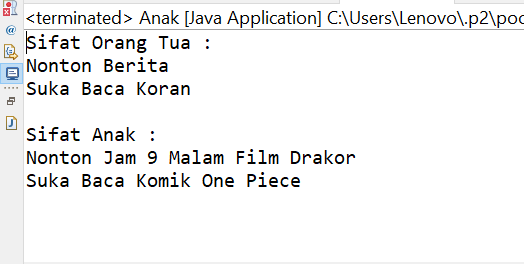
1. Mengevaluaisi method yang ada
2. Melakukan perbaikan penulisan kode method.
3. Kode program dan luaran
4. Screenshot/ Capture potongan kode dan hasil luaran setelah perbaikan

Beri komentar pada kode yang di Screenshot



1. Analisa luaran yang dihasilkan

Jawab:



Luaran telah sesuai dengan program yang disusun.   
Tipe data yang ditampilkan juga sudah sesuai dengan kebutuhan dan permintaan data.

**[No.1] Kesimpulan**

1. **Analisa**
2. Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
3. Permasalahan

* Penulisan method yang belum efisien

1. Algoritma

* Mengevaluaisi method yang ada
* Melakukan perbaikan penulisan kode method.

1. Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

Jawab:

Dasar pengambilan keputusan ini adalah agar mengefisienkan waktu eksekusi.

(Tuliskan singkat tentang pengalaman belajar, pemaknaan pengetahuan yang baru, tantangan yang dihadapi pada minggu tersebut. Ringkasan singkat dari semua soal, bukan per soal)

Jawab:

Pada minggu ini, pengalaman belajar saya adalah berusaha memahami tentang konsep - konsep baru dan menurut saya pribadi merasa itu adalah hal - hal yang cukup asing. Tantangan utama yang dihadapi adalah beradaptasi terhadap metode pembelajaran yang berbeda, serta kesulitan dalam mengaplikasikan teori ke dalam praktik. Namun, atas bantuan dari teman – teman membuat tugas ini menjadi lebih mudah dipahami.

Menurut saya pada soal 1 – 4 membuat Saya belajar bagaimana mendeklarasikan kelas dengan atribut dan method. Konsep constructor juga menjadi jelas; constructor digunakan untuk menginisialisasi objek saat dibuat.