**Template Lembar Kerja Individu dan Kelompok**

|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
|  |  |  |
| **[Nomor Soal] Identifikasi Masalah:** | | |
| 1. Uraikan permasalahan dan variabel 2. Rincikan sumber informasi yang relevan (buku / webpage) 3. Uraikan rancangan solusi yang diusulkan (jika ada). 4. Analisis susunan solusi, parameter solusi (jika ada). | | |
| **[Nomor Soal] Analisis dan Argumentasi** | | |
| 1. Uraikan rancangan solusi yang diusulkan. 2. Analisis solusi, kaitkan dengan permasalahan. | | |
| **[Nomor Soal] Penyusunan Algoritma dan Kode Program** | | |
| 1. Rancang desain solusi atau algoritma 2. Tuliskan kode program dan luaran 3. Beri komentar pada kode 4. Uraikan luaran yang dihasilkan 5. Screenshot/ Capture potongan kode dan hasil luaran | | |
| **[Nomor Soal] Kesimpulan** | | |
| 1. Analisa 2. Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program! 3. Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini? 4. Evaluasi 5. Apa konsekuensi dari skenario pemprograman ini? 6. Evaluasi input, proses, dan luaran yang dihasilkan! (jika ada) 7. Kreasi 8. Apakah ada pengetahuan baru yang dikembangkan dan konsep baru sebagai usulan solusi? 9. Konstruksikan hubungan antara variabel yang berbeda dengan konsep yang anda ketahui! (jika ada) | | |
|  | | |

Jawaban:

|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Cindy Aurora G1A024066** | **Kelas, Objek, Method** | **19 September 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)

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

Perbaikan line 6 dengan public Manusia(String nama, String rambut) {

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

1. Algoritma

a. memindahkan codingan ke online java compiler

b. menganalisa mana yang salah seperti line 6 harus diubah

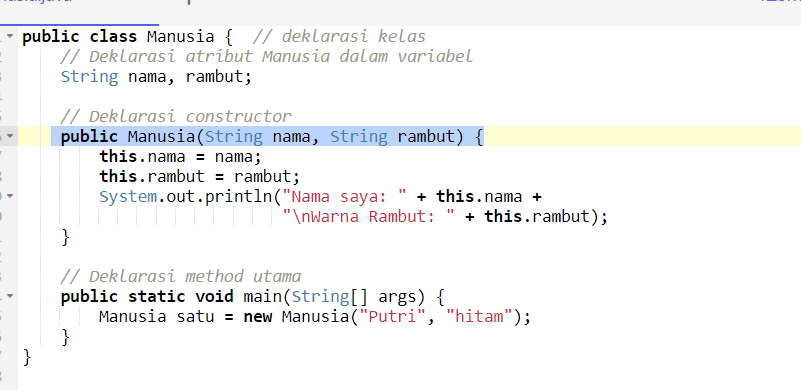
c. ubah line 6 dengan public Manusia(String nama, String rambut) {

d. lalu cetak hasil

e. run agar muncul hasil output

f. selesai

1. Kode program dan luaran



1. Screenshot/ Capture potongan kode dan hasil luaran
2. Analisa luaran yang dihasilkan   
   Contoh:

Permaslaahan awal ada di line 6 yang awalnya public Manusia1 (String nama) { , menjadi public Manusia(String nama, String rambut) {

**[No.1] Kesimpulan   
(PILIH SALAH SATU ANDA INGIN MEMBAHAS DENGAN CARA ANALISA/ EVALUASI / KREASI)**

1. **Analisa**

Pada awal, saya mengubah line 6 menjadi public Manusia(String nama, String rambut) {, lalu saya mengubah nama saya menjadi “Cindy”. Ketika dirun maka terlihat hasil seperti Output:

Nama saya: Cindy

Warna Rambut: hitam

**Refleksi**

Saya jadi paham tentang pembelajaran tersebut

Jawaban:

|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Cindy Aurora G1A024066** | **Kelas, Objek, Method** | **19 September 2024** |

**[No. 1] 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)

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

Pada line 3 terdapat kesalahan, saya mengganti kode program tersebut.

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

1. Algoritma

a. memindahkan codingan ke online java compiler

b. menganalisa mana yang salah

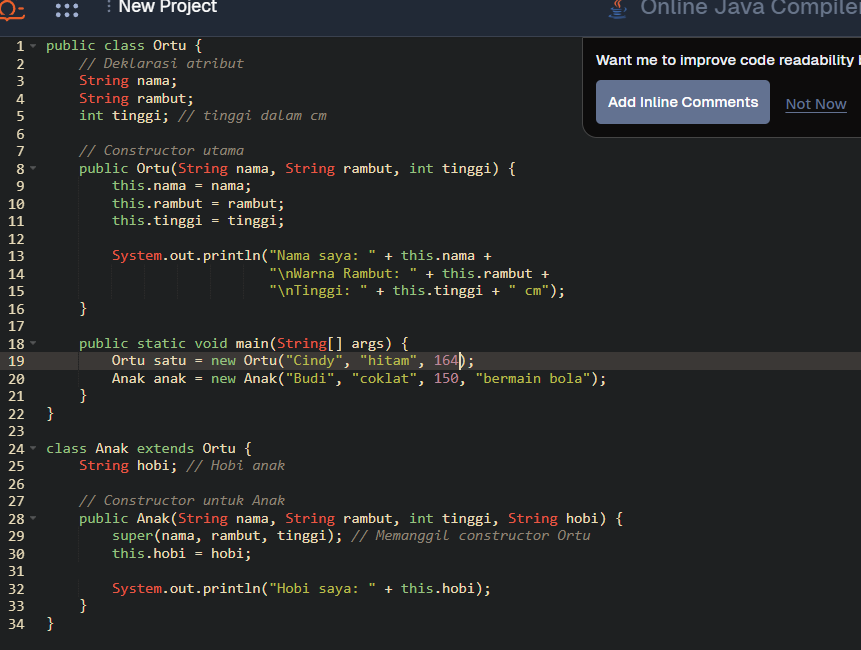
c. mengubah kode program

d. lalu cetak hasil

e. run agar muncul hasil output

f. selesai

1. Kode program dan luaran



1. Screenshot/ Capture potongan kode dan hasil luaran
2. Analisa luaran yang dihasilkan   
   Contoh:

a. **Kelas Anak**: Kelas ini mewarisi atribut dari kelas Ortu dan menambahkan atribut hobi.

b. **Constructor Overloaded**: Constructor Anak memanggil constructor dari Ortu dengan menggunakan super().

c. **Tampilan Informasi**: Saat objek dibuat, informasi mengenai nama, warna rambut, tinggi, dan hobi akan ditampilkan

**[No.1] Kesimpulan   
(PILIH SALAH SATU ANDA INGIN MEMBAHAS DENGAN CARA ANALISA/ EVALUASI / KREASI)**

1. **Analisa**

Saya menganalisis bahwa:

a. **Nama Constructor**: Constructor harus bernama sama dengan kelas (Ortu), dan didefinisikan dengan tanda kurung.

b. **Parameter Constructor**: Tambahkan parameter untuk nama, rambut, dan atribut lainnya (misal tinggi).

c. **Deklarasi Atribut**: Pastikan semua atribut (nama, rambut, tinggi) dideklarasikan.

Ketika di Run maka muncul hasil output seperti ini:

Nama saya: Putri

Warna Rambut: hitam

Tinggi: 160 cm

Nama saya: Budi

Warna Rambut: coklat

Tinggi: 150 cm

Hobi saya: bermain bola

**Refleksi**

Saya jadi paham tentang pembelajaran tersebut

Jawaban:

|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Cindy Aurora G1A024066** | **Kelas, Objek, Method** | **19 September 2024** |

**[No. 1] 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 1:**  
Manusia.java:6: error: invalid method declaration; return type required

public Manusia1(String nama, String rambut) {

^

Manusia.java:12: error: '(' expected

void sukaNonton {

^

Manusia.java:16: error: ';' expected

int sukaNonton {

^

Manusia.java:17: error: not a statement

episode\*durasi;

^

4 errors

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

Menurut saya ada penyebab eror tersebut seperti:

A. **Nama Constructor**: Constructor harus memiliki nama yang sama dengan kelas (Manusia), bukan Manusia1.

B. **Definisi Method**: Metode sukaNonton perlu didefinisikan dengan tanda kurung (), dan jika memiliki parameter, seharusnya menyertakan parameter tersebut.

C. **Return Type**: Metode sukaNonton yang kedua tidak memiliki tipe kembalian yang benar. Jika itu adalah metode yang mengembalikan nilai, perlu dituliskan sebagai int.

D. **Variabel yang Tidak Dideklarasikan**: Variabel film, episode, dan durasi tidak dideklarasikan dalam kelas.

E. **Menghitung Jumlah Jam**: Anda perlu mengubah logika pada metode sukaNonton untuk menghitung jumlah jam berdasarkan input.

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

1. Algoritma

a. memindahkan codingan ke online java compiler

b. menganalisa mana yang salah

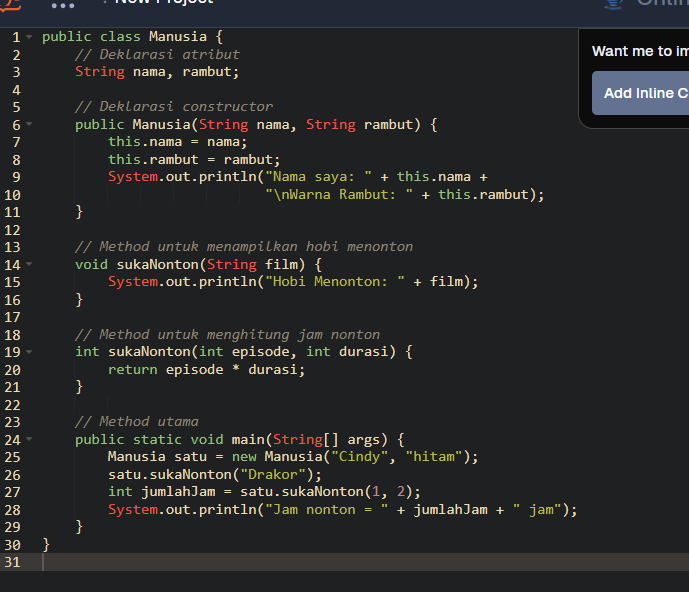
c. Mengubah dan menambah kode program agar tidak terjadi eror saat di Run.

d. lalu cetak hasil

e. run agar muncul hasil output

f. selesai

1. Kode program dan luaran



1. Screenshot/ Capture potongan kode dan hasil luaran
2. Analisa luaran yang dihasilkan   
   Contoh:

Permasalahan awal ada di

A. **Nama Constructor**: Constructor harus memiliki nama yang sama dengan kelas (Manusia), bukan Manusia1.

B. **Definisi Method**: Metode sukaNonton perlu didefinisikan dengan tanda kurung (), dan jika memiliki parameter, seharusnya menyertakan parameter tersebut.

C. **Return Type**: Metode sukaNonton yang kedua tidak memiliki tipe kembalian yang benar. Jika itu adalah metode yang mengembalikan nilai, perlu dituliskan sebagai int.

D. **Variabel yang Tidak Dideklarasikan**: Variabel film, episode, dan durasi tidak dideklarasikan dalam kelas.

E. **Menghitung Jumlah Jam**: Anda perlu mengubah logika pada metode sukaNonton untuk menghitung jumlah jam berdasarkan input.

**[No.1] Kesimpulan   
(PILIH SALAH SATU ANDA INGIN MEMBAHAS DENGAN CARA ANALISA/ EVALUASI / KREASI)**

1. **Analisa**

 **Constructor**: Diganti nama menjadi Manusia.

 **Method sukaNonton**: Diperbaiki untuk menerima parameter dan ditambahkan tanda kurung.

 **Method Menghitung Jam**: Menghitung dan mengembalikan jumlah jam berdasarkan episode dan durasi.

 **Penggunaan this**: Menandakan bahwa atribut di kelas sedang digunakan.

**Refleksi**

Saya jadi paham tentang pembelajaran tersebut

Jawaban:

|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Cindy Aurora G1A024066** | **Kelas, Objek, Method** | **19 September 2024** |

**[No. 1] 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

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

Perbaikan 

a. **Deklarasi nilaiU**: Menambahkan input untuk nilaiU sebelum mengevaluasi.

b. **Penggunaan Satu Scanner**: Hanya menggunakan satu objek Scanner untuk semua input.

c. **Menggunakan nextInt()**: Mengganti nextByte() dengan nextInt() untuk menghindari pembatasan nilai.

d. **Logika yang Jelas**: Struktur logika jika-else telah diperbaiki untuk memastikan output yang diharapkan muncul sesuai kondisi.

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

1. Algoritma

a. memindahkan codingan ke online java compiler

b. menganalisa mana yang salah

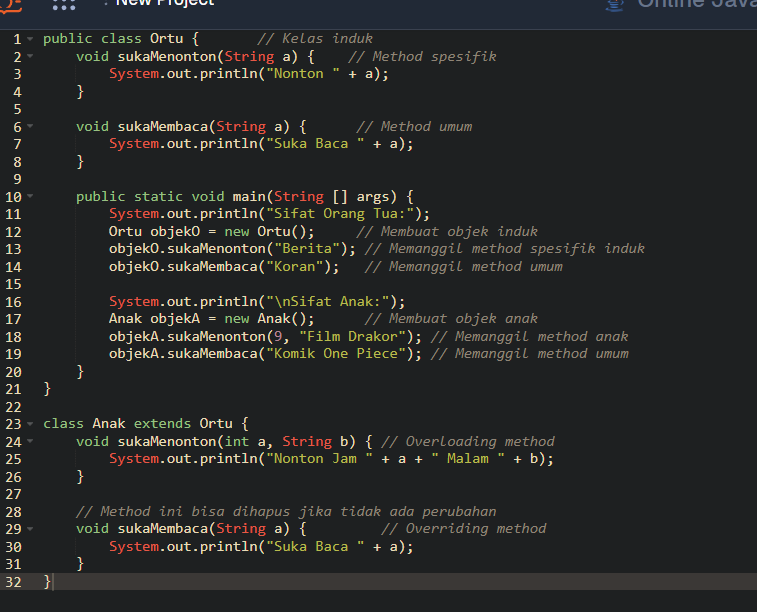
c. Mengubah dan menambahkan kode program agar bisa berjalan

d. lalu cetak hasil

e. run agar muncul hasil output

f. selesai

1. Kode program dan luaran



1. Screenshot/ Capture potongan kode dan hasil luaran
2. Analisa luaran yang dihasilkan   
   Contoh:

A. **Deklarasi nilaiU**: Menambahkan input untuk nilaiU sebelum mengevaluasi.

B. **Penggunaan Satu Scanner**: Hanya menggunakan satu objek Scanner untuk semua input.

C. **Menggunakan nextInt()**: Mengganti nextByte() dengan nextInt() untuk menghindari pembatasan nilai.

D. **Logika yang Jelas**: Struktur logika jika-else telah diperbaiki untuk memastikan output yang diharapkan muncul sesuai kondisi.

**[No.1] Kesimpulan   
(PILIH SALAH SATU ANDA INGIN MEMBAHAS DENGAN CARA ANALISA/ EVALUASI / KREASI)**

1. **Analisa**

 **Scanner**: Hanya perlu satu objek Scanner untuk membaca input dari pengguna. Menggunakan nextInt() agar dapat menangani nilai lebih besar dari byte.

 **Input Nilai**: Menambahkan input untuk nilaiU, nilaiT, dan nilaiQ agar semua nilai yang diperlukan diambil dari pengguna.

 **Logika Penentuan**: Menggunakan struktur if-else untuk memeriksa apakah nilaiU dan nilaiT masing-masing di atas 80, dan memberikan output yang sesuai.

 **Menutup Scanner**: Menggunakan scanner.close() untuk menutup objek Scanner setelah digunakan, menjaga kebersihan kode.

**Refleksi**

Saya jadi paham tentang pembelajaran tersebut