Lembar Kerja Individu

Nama & NPM	**Topik:**	**Tanggal:**
Rayhan Prabowo G1F024022	**Kelas,Objek,Method**	**19 September 2024**

[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
    a) Beri komentar pada kode
    b) Uraikan luaran yang dihasilkan
    c) Screenshot/ Capture potongan kode dan hasil luaran

[Nomor Soal] Kesimpulan

1) Analisa
    a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
    b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?
2) Evaluasi
    a) Apa konsekuensi dari skenario pemprograman ini?
    b) Evaluasi input, proses, dan luaran yang dihasilkan! (jika ada)
3) Kreasi
    a) Apakah ada pengetahuan baru yang dikembangkan dan konsep baru sebagai usulan solusi?
    b) Konstruksikan hubungan antara variabel yang berbeda dengan konsep yang anda ketahui! (jika ada)

[No. 1] Identifikasi Masalah:

1) Uraikan permasalahan dan variabel

Contoh 1:

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

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

Latihan 1:

1.1. Analisa ciri-ciri umum Kelas Manusia yang dapat menjadi
   a. atribut variabel, dan
   b. perilaku/ behavior untuk method!

[No.1] Analisis dan Argumentasi

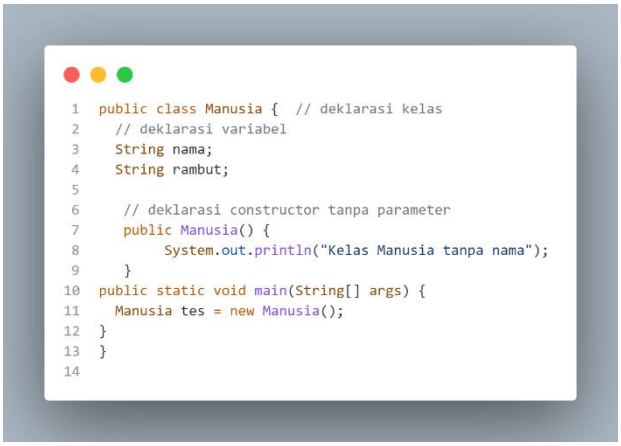
1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara :
   a. Inisialisasi variabel dalam constructor: Saat ini, variabel `nama` dan `rambut` tidak diinisialisasi. Kita bisa menginisialisasinya dengan nilai default.
   b. Tambahkan constructor berparameter: Untuk memungkinkan pembuatan objek Manusia dengan nilai awal yang spesifik.
   c. Tambahkan method toString(): Untuk representasi string yang lebih baik dari objek Manusia.
2) Alasan solusi ini karena tidak hanya untuk memperbaiki masalah langsung dalam kode , tetapi juga meningkatkan kualitas keseluruhan kelas, membuatnya lebih sesuai dengan prinsip-prinsip pemrograman yang baik dan lebih siap untuk pengembangan lebih lanjut.
3) Perbaikan kode program dengan cara :
   a. Ditambahkan constructor dengan parameter untuk memungkinkan inisialisasi objek dengan nilai awal.
   b. Ditambahkan method toString() untuk memberikan representasi string yang lebih bermakna dari objek
   c. Menambahkan komentar lebih banyak untuk meningkatkan kejelasan kode.
   d. Variabel (nama dan rambut) diubah menjadi private untuk encapsulation yang lebih baik.

[No.1] Penyusunan Algoritma dan Kode Program

1) Algoritma
   Algoritma adalah langkah-langkah penyelesaian masalah.
   Misalkan algoritma membuat program dengan variabel constructor:
   (a) Mulai
   (b) Deklarasikan variabel nama dan rambut dengan tipe data string.
   (c) Buat objek baru dari kelas manusia dengan nama `tes`
   (d) Panggil constructor manusia
   (e) Cetak “Kelas Manusia tanpa nama” ke konsol
   (f) Selesai.

2) Kode program dan luaran

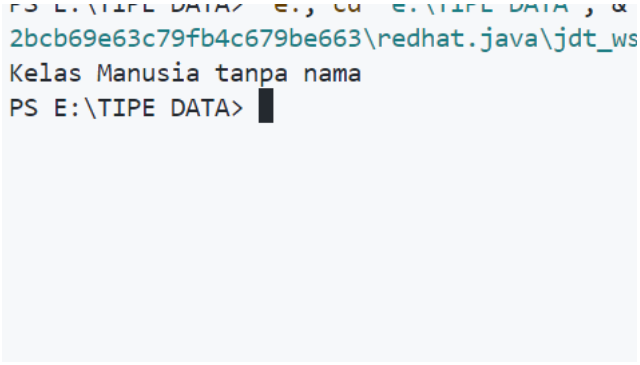
a) Kode program



```java
public class Manusia {  // deklarasi kelas
  // deklarasi variabel
  String nama;
  String rambut;

   // deklarasi constructor tanpa parameter
   public Manusia() {
        System.out.println("Kelas Manusia tanpa nama");
   }
public static void main(String[] args) {
  Manusia tes = new Manusia();
}
}
```

b) Luaran



```
2bcb69e63c79fb4c679be663\redhat.java\jdt_ws
Kelas Manusia tanpa nama
PS E:\TIPE DATA>
```

c) Analisa luaran yang dihasilkan

Program diatas berfungsi sebagai demonstrasi dasar dari pembuatan kelas dan objek dalam Java. Luarannya hanya berupa satu baris teks yang dicetak saat objek dibuat. Meskipun program berjalan tanpa error, ia tidak melakukan operasi yang berarti terhadap data (variabel instance) yang telah dideklarasikan.

[No.1] Kesimpulan

1) Analisa

a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!

b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

Jawab :

a) Analisa ciri umum kelas manusia :

a. Atribut variabel (Karakteristik)

i. Nama(seperti dalam contoh)
ii. Rambut (seperti dalam contoh)
iii. Umur
iv. Tinggi
v. Berat
vi. Jenis kelamin
vii. Alamat
viii. Pekerjaan
ix. Status pernikahan

x. Golongan darah
xi. No identitas(NIK)
xii. Tanggal lahir
xiii. Warna kulit
xiv. Bahasa utama
xv. Kewarganegaraan

b. Perilaku untuk method
i. Berbicara (string pesan)
ii. Berjalan (int jarak)
iii. Makan (string makanan)
iv. Tidur (int durasi)
v. Bekerja (string pekerjaan)
vi. Belajar (string subjek)
vii. Olahraga (string jenisolahraga)
viii. Bernyanyi (String lagu) dll.

b) Penjelasan
a. Atribut variabel mewakili karakteristik atau properti yang dimiliki oleh setiap manusia.Contoh nya (nama,rambut). Dan yang lain ditambahkan berdasarkan ciri ciri umum manusia.
b. Perilaku untuk method mewakili Tindakan atau kemampuan yang dapat dilakukan oleh manusia. Ini bisa berupa Tindakan fisik (seperti berjalan,makan).

c) Dasar pengambilan keputusan saya untuk kasus ini adalah dengan menambahkan atribut dan method ini, kelas Manusia akan menjadi lebih komprehensif dan realitis dalam merepresentasikan seorang manusia dalam konteks pemrograman berorientasi objek.

[No. 2] Identifikasi Masalah:

1) Uraikan permasalahan dan variabel

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

```
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("Putri", "hitam");
     }
}
```

Luaran 2:

```
Nama saya : Putri
Warna Rambut : hitam
```

Latihan 2:

2.1. Susun kembali kode di contoh 2 dengan menambahkan data ciri-ciri Anda di dalam variabel constructor!

2.2. Apabila nanti Anda akan memiliki keturunan, analisa sifat (atribut), constructor, dan perilaku positif (behavior) apa yang akan diturunkan?

[No.2] Analisis dan Argumentasi

1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara Menambahkan variabel warna kulit,berat badan badan,tinggi badan, dan umur.
2) Alasan solusi ini karena pada soal yang diberikan kita diinstruksikan untuk menambahkan ciri ciri kedalam variabel constructor yang telah dibuat pada contoh 2
3) Penambahan kode program dengan cara :

```
public  class operator {

  //deklarasi constructor
  public operator(String nama, String rambut, String kulit, String
tinggi, String berat, String umur)  {
    //nama dan rambut adalah variabel constructor
    System.out.println(" Nama saya : "+ nama + "\n Warna Rambut : " +
rambut + "\n Warna kulit : " + kulit + "\n Tinggi : " + tinggi + "\n
Berat : " + berat + "\n Umur : " + umur);

  }

  public static void main (String[] args) {
    operator saya = new operator("Rayhan", "hitam", "hitam", "165",
"45", "17");
  }
  }
```

[No.2] Penyusunan Algoritma dan Kode Program

1) Algoritma
   Algoritma adalah langkah-langkah penyelesaian masalah.
   Algoritma program variabel constructor:
   (a) Mulai.
   (b) Definisikan kelas operator
   (c) Deklarasikan konstruktor dengan rincian
      a. Nama (String)
      b. Rambut (String)
      c. Kulit (String)
      d. Tinggi (String)
      e. Berat (String)
      f. Umur(String)
   (d) Cetak informasi menggunakan System.out.println() dengan format: "Nama saya : [nama] Warna Rambut : [rambut] Warna kulit : [kulit] Tinggi : [tinggi] Berat : [berat] Umur : [umur]"
   (e) Selesai.

2) Kode program dan luaran

a) Kode Program

```
1  package operator;
2
3  public  class operator {
4
5       //deklarasi constructor
6       public operator(String nama, String rambut, String kulit, String tinggi, String berat, String umur)  {
7         //nama dan rambut adalah variabel constructor
8         System.out.println(" Nama saya : "+ nama + "\n Warna Rambut : " + rambut + "\n Warna kulit : " + kulit +
9                 "\n Tinggi : " + tinggi + "\n Berat : " + berat + "\n Umur : " + umur);
10
11      }
12
13
14      public static void main (String[] args) {
15          operator saya = new operator("Rayhan", "hitam", "hitam", "165", "45", "17");
16      }}
```

b) Luaran

```
Nama saya : Rayhan
Warna Rambut : hitam
Warna kulit : hitam
Tinggi : 165
Berat : 45
Umur : 17
```

c) Analisa luaran yang dihasilkan
Luaran yang dihasilkan sudah sesuai dengan data yang diminta pada soal,dengan menunjukan ciri ciri di dalam variabel constructor.

[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?

Jawab :

a) Analisa sifat (atribut),constructor, dan perilaku positif(behavior) yang dapat diturunkan kepada keturunan :
   a. Atribut (sifat) ,meliputi :
      i. Nama
      ii. Warna rambut
      iii. Warna mata
      iv. Tinggi badan
      v. Golongan darah
   b. Perilaku positif (behavior) :
      i. Belajar() – Metode untuk menggambarkan keinginan belajar
      ii. MenghormatiOrangTua() – Metode yang menunjukan sikap hormat
      iii. bekerjaKeras() - Metode yang menggambarkan etos kerja
      iv. berpikirKritis() – Metode untuk mengembangkan pemikiran kritis

b) Dasar pengambilan keputusan Analisa diatas adalah dalam membuat sebuah kelas anak yang mewarisi sifat – sifat dan perilaku positif dari orangtua maka harus ada Atribut(sifat) dan

perilaku positif(behavour) untuk menandakan bahwa benar dengan ciri – ciri dan perilaku keturunan tersebut cocok dengan orang tuanya

[No. 3] Identifikasi Masalah:

1) Uraikan permasalahan dan variabel

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(String film) {
        System.out.println(" Hobi Menonton : " + film);
    }

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

Luaran 3:

```
Nama saya : Putri
Warna Rambut : hitam
Hobi Menonton : Drakor
```

Latihan 3:

3.1. Analisa perbedaan deklarasi constructor, method, dan method utama!
3.2. Tentukan kapan Anda perlu menggunakan constructor dan method?
3.3. Uraikan perbedaan berikut:
   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 :
   a. Nama konstruktor diubah dari manusia1 menjadi manusia agar sesuai dengan nama kelas
   b. Variabel instance nama dan rambut diinisialisasi didalam konstruktor menggunakan keyword this untuk membedakan variabel instance dan parameter
   c. Konstruktor sekarang menggunakan variabel instance yang telah diinisialisasi untuk mencetak informasi
   d. Modifier private ditambahkan ke variabel instance untuk enkapsulasi yang lebih baik
   e. Metode “sukaNonton” tidak diubah karena benar
2) Alasan solusi ini karena dengan menerapkan solusi ini , kode menjadi lebih mudah dipahami, dan lebih sesuai dengan prinsip prinsip OOP.Hal ini penting tidak hanya untuk fungsi program saat ini, tetapi juga untuk pemeliharaan dan untuk pengembangan.
3) Modifikasi kode program dengan cara menganti variabel instance nama dan rambut menjadi nama asli dan warna rambut asli, jelasnya berikut kode program yang telah di modifikasi:

```
package Manusia;

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

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

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

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

[No.3] Penyusunan Algoritma dan Kode Program

1) Algoritma
   Algoritma adalah langkah-langkah penyelesaian masalah.
   Algoritma membuat program dengan method:
   (a) Mulai
   (b) Definisikan kelas manusia
   (c) Deklarasikan atribut nama(string) dan rambut (string)
   (d) Buat konstruktor manusia dengan parameter nama dan rambut
   (e) Cetak nama dan warna rambut
   (f) Buat method sukaNonton dengan parameter film(string)
   (g) Cetak hobi menonton film
   (h) Buat method main
   (i) Buat objek manusia Bernama ‘satu’ dengan nama “Rayhan” dan warna rambut “hitam”
   (j) Panggil method sukaNonton untuk objek ‘satu’ dengan parameter “Smackdown”
   (k) Panggil konstruktor manusia, cetak nama dan warna rambut
   (l) Panggil method sukaNonton, cetak hobi menonton
   (m) Selesai.

2) Kode program dan luaran

a) Kode program

```
1  package Manusia;
2
3  public class Manusia {
4      //deklarasi atribut Manusia dalam variabel
5      String nama, rambut;
6
7      //deklarasi constructor
8      public Manusia(String nama, String rambut) {
9              System.out.println(" Nama : "+ nama +
10             "\n Warna Rambut : " + rambut);
11     }
12
13     //deklarasi method
14     void sukaNonton(String film) {
15         System.out.println(" Hobi Menonton : " + film);
16     }
17
18     //deklarasi method utama
19     public static void main( String[] args) {
20             Manusia satu = new Manusia("Rayhan", "hitam");
21             satu.sukaNonton("Smackdown");
22     }
23 }
```

b) Luaran

```
Nama : Rayhan
Warna Rambut : hitam
Hobi Menonton : Smackdown
```

c) Analisa luaran yang dihasilkan

Luaran sudah sesuai dengan program yang disusun. Dengan mendemonstrasikan pembuatan objek sederhana dengan atribut dan method,serta penggunaan konstruktor untuk inisialisasi ke luaran.

[No.3] 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:

a) Analisa perbedaan deklarasi constructor,method, dan method utama :

a. Constructor :

- Nama constructor sama dengan nama kelas
- Tidak memiliki tipe pengembalian
- Digunakan untuk menginisialisasi objek

b. Method :

- Memiliki tipe pengembalian (void contohnya)
- Dapat dipanggil pada objek kelas

- Digunakan untuk melakukan operasi tertentu

c. Method utama (main)
    - Selalu dideklarasikan sebagai `public static void main(String[] args)`
    - merupakan titik masuk program java
    - dijalankan saat program dimulai

b) Kapan perlu menggunakan constructor dan method:
    a. Constructor digunakan Ketika:
        - Ingin menginisialisasi objek saat pembuatan
        - Perlu mengatur nilai awal atribut objek
        - Ingin melakukan operasi tertentu saat objek dibuat
    b. Method digunakan Ketika :
        - Ingin menambahkan fungsionalitas pada objek
        - Perlu melakukan operasi yang dapat dipanggil berualang kali
        - Ingin mengorganisir kode menjadi blok blok yang dapat digunakan Kembali.

c) Perbedaan constructor overloading dan overleading :
    a. Constructor overloading : valid dan sering digunakan dalam java untuk menyediakan berbagai cara menginisialisasi objek.
    b. Constructor overriding: tidak ada dalam java. Perlu di konfirmasi bahwa mungkin tidak ada constructor overriding dalam java.constructor tidak dapat di override karena :
        - Constructor tidak diwariskan ke kelas anak
        - Constructor selalu memiliki nama yang sama dengan nama kelas, sehingga tidak mungkin untuk mengganti (override) constructor kelas induk di kelas anak.

d) Perbedaan method overloading dan method overriding
    a. Method overloading :
        - Terjadi dalam kelas yang sama.
        - Metode memiliki nama yang sama tetapi parameter berbeda (jumlah atau tipe data)
        - Memungkinkan fleksibilitas dalam pemanggilan metode:
    b. Method overriding :
        - Terjadi antara kelas insduk dan kelas anak
        - Metode di kelas memiliki nama dan parameter yang sama dengan metode dikelas untuk.
        - Digunakan untuk memberikan implementasi khusus dari metode kelas induk dikelas anak.

e) Perbedaan method yang mengembalikan nilai dan tidak mengembalikan nilai :
    a. Method yang mengembalikan nilai :
        - Memiliki tipe pengembalian selain void
        - Menggunakan keyword return untuk mengembalikan nilai
        - Nilai yag dikembalikan harus sesuai dengan tipe pengembalian yang dideklarasikan
    b. Method void (tidak mengembalikan nilai) :
        - Menggunakan keyword void sebagai tipe pengembalian

- Tidak menggunakan return untuk mengembalikan nilai (bisa menggunakan return; untuk keluar method lebih awal).
- Biasanya digunakan untuk melakukan aksi tanpa perlu mengembalikan hasil.

[No. 4] Identifikasi Masalah:

1) Uraikan permasalahan dan variabel
   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. Bandingkan method yang dimiliki `class Anak extends Ortu` dengan method di `class Ortu`!
4.2. Ubahlah Contoh 4 dengan menambahkan objek anak dengan method yang berbeda!

[No.4] Analisis dan Argumentasi

1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara memodifikasi kode diatas dengan melakukan perubahan dan penambahan yaitu
   a. Kelas AnakKedua
      - Me-override method sukaMenonton(String a) dengan implementasi yang berbeda.
      - Menambahkan method baru sukaBermain(String a) yang tidak ada dikelas induk.
   b. Dalam method main :
      - Menambahkan objek dari kelas Anakkedua.
      - Memanggil method sukaMenonton dan sukaBermain dari objek AnakKedua.
2) Alasan solusi ini karena untuk memberikan gambaran tentang konsep konsep penting dalam pemrograman berorientasi objek di java,untuk tetap menjaga kode agar mudah dipahami dan diikuti _____
3) Perbaikan kode program dengan cara Menambahkan objek anak dengan mothod yang berbeda jelasnya perhatikan kode program yang sudah diubah dibawah ini :

```
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
```

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

        System.out.println("\nSifat 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

        System.out.println("\nSifat Anak Kedua :");
        AnakKedua objekA2 = new AnakKedua();    //memanggil objek anak
kedua
        objekA2.sukaMenonton("FilmHorror");        //memanggil method
yang di-override
        objekA2.sukaBermain("Game Online");   //memanggil method baru
    }
}

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

    @Override
 void sukaMenonton(String a) {
        // method induk spesifik yang di-override
        System.out.println("Nonton " + a + " sambil makan popcorn");
    }

    @Override
    void sukaMembaca(String a) {
        // method induk umum yang di-override
        System.out.println("Suka Baca " + a + " sambil tiduran");
    }
}

class AnakKedua extends Ortu {
    @Override
    void sukaMenonton(String a) {
        // method induk spesifik yang di-override
        System.out.println("Nonton " + a + " sambil belajar");
    }

    void sukaBermain(String a) {
        // method baru yang tidak ada di kelas induk
        System.out.println("Suka Bermain " + a);
    }
```

```
}
```

[No.4] Penyusunan Algoritma dan Kode Program

1) Algoritma
   Algoritma adalah langkah-langkah penyelesaian masalah.
   Algoritma program dengan class extends:
   (a) Mulai
   (b) Definisikan kelas ortu :
   - Buat method sukaMenonton (String a)
   - Buat method sukaMembaca (String a)

   (c) Definisikan kelas anak yang mewarisi ortu :
   - Override method sukaMenonton (String a)
   - Override method sukaMembaca (String a)
   - Tambahkan method baru sukaMenonton (int a, String b)

   (d) Definisikan kelas AnakKedua yang mewarisi Ortu :
   - Override methd sukaMenonton (String a)
   - Tambahkan method baru sukaBermain (String a)

   (e) Didalam method main :
   - Tampilkan “Sifat Orang Tua”
   - Buat objek ortu
   - Panggil sukaMembaca (“Koran”) dari objek ortu
   - Tampilkan “\nSifat Anak :”
   - Buat objek anak
   - Panggil sukaMenonton(“Film Drakor”) dari objek anak
   - Tampilkan “\nSifat Anak Kedua :”
   - Buat objek Anak Kedua
   - Panggil sukaMenonton(9, “FimHorror”) dari objek AnakKedua
   - panggil sukaBermain(“Game Online”) dari objek anak kedua

   (f) Selesai.

2) Kode program dan luaran

a) Kode Program

```java
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("\nSifat 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

        System.out.println("\nSifat Anak Kedua :");
        AnakKedua objekA2 = new AnakKedua();    //memanggil objek anak kedua
        objekA2.sukaMenonton("FilmHorror");        //memanggil method yang di-override
        objekA2.sukaBermain("Game Online");   //memanggil method baru
    }
}

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

    @Override
    void sukaMenonton(String a) {
        // method induk spesifik yang di-override
        System.out.println("Nonton " + a + " sambil makan popcorn");
    }

    @Override
    void sukaMembaca(String a) {
        // method induk umum yang di-override
        System.out.println("Suka Baca " + a + " sambil tiduran");
    }
}

class AnakKedua extends Ortu {
    @Override
    void sukaMenonton(String a) {
        // method induk spesifik yang di-override
        System.out.println("Nonton " + a + " sambil belajar");
    }

    void sukaBermain(String a) {
        // method baru yang tidak ada di kelas induk
        System.out.println("Suka Bermain " + a);
    }
}
```

b) Luaran

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

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

Sifat Anak Kedua :
Nonton FilmHorror sambil belajar
Suka Bermain Game Online
```

c) Analisa luaran yang dihasilkan

Luaran sudah sesuai dengan program yang disusun. Dengan menambahkan kode objek anak dengan method yang berbeda.

[No.4] Kesimpulan

1) Evaluasi

a) Apa konsekuensi/dampak dari kode program yang dibuat?
b) Evaluasi input program, proses perhitungan, dan luaran yang dihasilkan! (jika ada)

Jawab:

a) Dari kasus diatas saya mengevalusi perbandingan method yang dimiliki class Anak extends Ortu dengan method di class ortu;
    a. Method di class Ortu :
        i. sukaMenonton(String a)
        ii. sukaMembaca (String a)
    b. Method di class Anak :
        i. sukaMenonton(String a) di-override
        ii. sukaMembaca (String a) di-override
        iii. sukaMenonton (int a, String a) di-override
    c. Kesimpulan dari perbandingan method yang dimiliki class anak extends ortu dengan method class ortu yaitu jika class anak extends ortu di-override sedangkang di class ortu tidak di override atau tidak dengan metode overriding.
b) Program yang saya buat ini memberikan dampak positif dalam hal demonstrasi kosep OOP,peningkatan reusabilitas kode, dan fleksibilitas dalam pengembangan.

Refleksi

Saya menyelesaikan tugas ini dengan hanya satu malam,karena tugas ini termasuk tugas yang dikategorikan tugas dengan tingkat kesulitan menengah.Saya memberikan usaha yang terbaik saat mengerjakan tugas ini,banyak hal hal baru yang saya pelajari saat mengerjakan tugas ini . Saya bisa meningkatkan kemampuan saya saat mengerjakan tugas ini dengan cara mengganti variabel pada contoh data atau ciri ciri yang di instruksikan. Pada materi yang mendatang saya harap saya bisa mendapatkan materi yang lebih sulit karena jika saya mendapatkan materi yang lebih sulit maka akan lebih banyak ilmu yang saya dapatkan dari tugas tersebut.