

Lembar Kerja Individu

Nama & NPM	Topik:	Tanggal:
Anggita Afriyani G1F024011	Kelas, Objek, Method	17 September 2024

[No. 1] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variabel

Contoh:

Tuliskan kembali soal:

```
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:

- 1) `nama` : Merupakan atribut yang menyimpan nama dari objek `manusia`.

- 2) `rambut` : Merupakan atribut yang menyimpan informasi tentang rambut dari objek . `Manusia`.

- b. perilaku/ behavior untuk method!

Konstruktor : Metode ini akan dipanggil saat objek dari kelas dibuat dan mencetak pesan "Kelas Manusia tanpa nama".

Pada soal masih terdapat kesalahan "variabel `nama` dan `rambut`".

[No.1] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara penambahan parameter pada konstruktor untuk menginisialisasi atribut `nama` dan `rambut`.
- 2) Alasan solusi ini karena beberapa variabel tidak memiliki nilai awal sehingga ketika objek dibuat nilai tersebut tetap null.
- 3) Perbaiki kode program dengan cara penambahan konstruktor dengan parameter untuk menginisialisasi atribut saat objek dibuat.

[No.1] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
Algoritma adalah langkah-langkah penyelesaian masalah.
Misalkan algoritma memasak mi instan:
 - (a) Mulai program
 - (b) Deklarasi kelas manusia
 - (c) Buat konstruktor tanpa parameter
 - (d) Buat konstruktor dengan parameter
 - (e) Buat metode untuk menampilkan informasi
 - (f) Deklarasi kelas utama
 - (g) Metode utama

(h) Hasil output.

2) Kode program dan luaran

```
1 public class Manusia {
2     private String nama;
3     private String rambut;
4
5     public Manusia() {
6         System.out.println("Kelas Manusia tanpa nama");
7     }
8
9     public Manusia(String nama, String rambut) {
10        this.nama = nama;
11        this.ambut = rambut;
12        System.out.println("Kelas Manusia dengan nama: " + this.nama + " dan rambut: " + this.ambut);
13    }
14
15    public void tampilkanInfo() {
16        if (nama != null && rambut != null) {
17            System.out.println("Nama: " + nama);
18            System.out.println("Rambut: " + rambut);
19        } else {
20            System.out.println("Nama dan rambut belum diinisialisasi");
21        }
22    }
23
24    public String getNama() {
25        return nama;
26    }
27
28    public String getRambut() {
29        return rambut;
30    }
31 }
32
33 class Main {
34     public static void main(String[] args) {
35         Manusia manusia1 = new Manusia();
36         manusia1.tampilkanInfo();
37
38         Manusia manusia2 = new Manusia("Ali", "Hitam");
39         manusia2.tampilkanInfo();
40
41         Manusia manusia3 = new Manusia("Budi", "Coklat");
42         manusia3.tampilkanInfo();
43
44         Manusia manusia4 = new Manusia("Siti", "Pirang");
45         manusia4.tampilkanInfo();
46     }
47 }
```

a) Screenshot/ Capture potongan kode dan hasil luaran

Output Generated Files

```
Kelas Manusia tanpa nama
Nama dan rambut belum diinisialisasi
Kelas Manusia dengan nama: Ali dan rambut: Hitam
Nama: Ali
Rambut: Hitam
Kelas Manusia dengan nama: Budi dan rambut: Coklat
Nama: Budi
Rambut: Coklat
Kelas Manusia dengan nama: Siti dan rambut: Pirang
Nama: Siti
Rambut: Pirang
```

b) Analisa luaran yang dihasilkan

Hasil luaran telah sesuai dengan program yang disusun. Tampilan tipe data telah sesuai dengan kebutuhan data.

Luaran sudah sesuai dengan program yang disusun.

Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

[No.1] Kesimpulan

1) Analisa

- Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!

Penggunaan bentuk kelas public karena kelas harus dapat diakses dari luar paket. Lalu, dengan penambahan konstruktor dengan parameter karena diperlukannya inisiasi atribut untuk menghindari nilai default null.

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

Alasan dasar pengambilan keputusan ini ialah untuk memastikan bahwa setiap objek memiliki nilai yang jelas untuk atribut sehingga dapat digunakan dalam konteks aplikasi yang lebih besar.

[No. 2] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variabel

Contoh:

Tuliskan kembali soal:

```
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!

Dengan menambahkan beberapa atribut baru ke dalam konstruktor.

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

Sifat (atribut):

- `String nama` : Nama orang tua
- `String rambut` : warna rambut orang tua
- `int umur` : umur orang tua
- `String pekerjaan` : pekerjaan orang tua

Konstruktor yang diturunkan ke kelas anak harus mencakup semua atribut yang relevan dari kelas. Misalnya jika kita membuat kelas, dapat mewarisi atribut dan menambahkannya sesuai kebutuhan.

Perilaku positif

- Metode penampilan informasi
- Metode untuk menghitung usia

[No.2] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan menambah lebih banyak atribut ke dalam kelas untuk memberikan informasi lengkap terkait objek.
- 2) Alasan solusi ini karena kita jadi dapat memberikan konteks yang lebih baik tentang karakteristik orang tua.
- 3) Perbaiki kode program dengan cara memperkenalkan atribut baru dalam konstruktor dan mendeklarasikan variabel sebagai privat untuk menjaga enkapsulasi.

[No.2] Penyusunan Algoritma dan Kode Program

1) Algoritma

- (a) Deklarasi kelas `Ortu`
- (b) Tambahkan atribut untuk nama, rambut, umur, dan pekerjaan.
- (c) Buat konstruktor untuk menginisialisasi atribut.
- (d) Buat metode untuk menampilkan informasi
- (e) Buat objek dari kelas dengan data yang relevan di dalam metode.

2) Kode program dan luaran

```
1 public class Ortu {  
2     // Deklarasi atribut  
3     private String nama;  
4     private String rambut;  
5     private int umur;  
6     private String pekerjaan;  
7  
8     // Deklarasi constructor  
9     public Ortu(String nama, String rambut, int umur, String pekerjaan) {  
10         this.nama = nama;  
11         this.rambut = rambut;  
12         this.umur = umur;  
13         this.pekerjaan = pekerjaan;  
14  
15         System.out.println("Nama saya : " + nama +  
16                             "\nWarna Rambut : " + rambut +  
17                             "\nUmur : " + umur +  
18                             "\nPekerjaan : " + pekerjaan);  
19     }  
20  
21     public static void main(String[] args) {  
22         Ortu satu = new Ortu("Putri", "hitam", 40, "Dokter");  
23     }  
24 }
```

Output Generated Files

```
Nama saya : Putri  
Warna Rambut : hitam  
Umur : 40  
Pekerjaan : Dokter
```

a) Screenshot/ Capture potongan kode dan hasil luaran

b) Analisa luaran yang dihasilkan

Contoh:

Luaran sudah sesuai dengan program yang disusun.

Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

luaran telah sesuai dengan program, tipe data sudah sesuai permintaan, dan program berhasil menampilkan informasi lengkap tentang `Ortu`

[No.2] Kesimpulan

1) Analisa

- a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
Kesimpulannya menunjukkan bahwa penambahan atribut baru memperkaya informasi objek.
- b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?
Penggunaan kelas public karena diperlkan aksesibilitas dari luar kelas serta perbaikan program dengan penambahan atribut baru karena di struktur java memerlukan penggunaan enkapsulasi untuk melindungi data.

[No. 3] Identifikasi Masalah:

1) Uraikan permasalahan dan variabel

Contoh:

Tuliskan kembali soal:

```
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!

a) constructor overloading dan overriding

merupakan metode khusus yang digunakan untuk menginisialisasi objek ketika dibuat. Nama constructor harus sama dengan nama kelas.

Contoh: public Manusia(String nama, String rambut). Berfungsi untuk menginisialisasi atribut objek dan melakukan tindakan yang diperlukan saat objek dibuat.

method overloading, dan method overriding

b) Method adalah fungsi yang didefinisikan dalam kelas yang dapat melakukan tindakan tertentu. Method dapat memiliki parameter dan dapat mengembalikan nilai. Berfungsi menyediakan perilaku atau aksi yang dapat dilakukan oleh objek.

c) method yang mengembalikan nilai dan method tidak mengembalikan nilai merupakan titik awal eksekusi program java. Nama method ini selalu `main` dengan parameter `String [] args`. Dan berfungsi menjalankan program dengan memanggil metode lain.

3.2. Tentukan kapan Anda perlu menggunakan constructor dan method?

Konstruktor ketika ingin menginisialisasi atribut objek saat objek dibuat, sedangkan menggunakan method ketika untuk melakukan tindakan tertentu setelah objek dibuat.

3.3. Uraikan perbedaan berikut:

- Overloading: Memungkinkan beberapa constructor dengan nama yang sama tetapi dengan parameter yang berbeda dalam satu kelas.

- Overriding: Tidak berlaku untuk constructor karena constructor tidak dapat diwarisi.

b) Method Overloading dan Method Overriding:

- Overloading: Memungkinkan beberapa method dengan nama yang sama tetapi dengan parameter yang berbeda dalam satu kelas.

- Overriding: Memungkinkan subclass untuk menyediakan implementasi spesifik dari method yang sudah didefinisikan di superclass.

c) Method yang Mengembalikan Nilai dan Method Tidak Mengembalikan Nilai:

1. Method dideklarasikan dengan tipe data tertentu (misalnya int, String) dan menggunakan pernyataan return untuk mengembalikan nilai.

2. Dideklarasikan dengan tipe void dan tidak menggunakan pernyataan return untuk mengembalikan nilai.

[No.3] Analisis dan Argumentasi

- Saya mengusulkan permasalahan ini dapat diatasi dengan cara Memperbaiki deklarasi constructor agar sesuai dengan nama kelas.
- Alasan solusi ini karena constructor harus memiliki nama yang sama dengan kelas agar dapat diidentifikasi oleh Java saat membuat objek.
- Perbaiki kode program dengan cara mengubah nama konstruktor `manusia1` menjadi `manusia`

[No.3] Penyusunan Algoritma dan Kode Program

1) Algoritma

(a) Deklarasikan kelas Manusia.

(b) Tambahkan atribut untuk nama dan rambut.

(c) Buat constructor untuk menginisialisasi atribut.

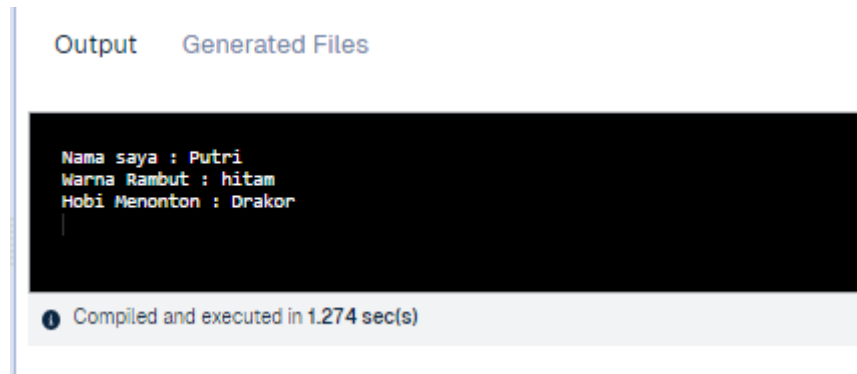
(d) Buat metode untuk menampilkan hobi menonton.

(e) Dalam metode utama, buat objek dari kelas Manusia dan panggil metode hobi menonton.

2) Kode program dan luaran

```
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) { // Perbaiki nama constructor  
7         System.out.println("Nama saya : " + nama +  
8             "\nWarna Rambut : " + rambut);  
9     }  
10  
11     // Deklarasi method  
12 - void sukaNonton(String film) {  
13         System.out.println("Hobi Menonton : " + film);  
14     }  
15  
16     // Deklarasi method utama  
17 - public static void main(String[] args) {  
18         Manusia satu = new Manusia("Putri", "hitam");  
19         satu.sukaNonton("Drakor");  
20     }  
21 }
```

a) Screenshot/ Capture potongan kode dan hasil luaran



The screenshot shows a Java IDE interface with two tabs: 'Output' and 'Generated Files'. The 'Output' tab is active, displaying the following text in a monospaced font: 'Nama saya : Putri', 'Warna Rambut : hitam', and 'Hobi Menonton : Drakor'. Below the output, a status bar indicates 'Compiled and executed in 1.274 sec(s)'.

- b) Analisa luaran yang dihasilkan
Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.
Program berhasil menampilkan informasi lengkap tentang objek Manusia serta hobi menontonnya.

[No.3] Kesimpulan

- 1) **Analisa**
 - a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program! a)
Kesimpulan berdasarkan permasalahan, algoritma, dan kode program menunjukkan bahwa perbaikan pada nama constructor sangat penting agar program dapat berjalan tanpa kesalahan.
 - b) Dasar alasan pengambilan keputusan saya untuk kasus ini adalah:
Pada program itu saya menggunakan bentuk kelas public karena diperlukan aksesibilitas dari luar kelas.
Perbaikan program dengan menambahkan perubahan pada nama constructor karena struktur Java mengharuskan bahwa constructor harus memiliki nama yang sama dengan kelasnya.
 - c) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

[No. 4] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variabel

Contoh:

Tuliskan kembali soal:

```
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!

- Class Ortu:

void sukaMenonton(String a) : Method spesifik untuk menonton.
 void sukaMembaca(String a) : Method umum untuk membaca.

- Class Anak:

void sukaMenonton(int a, String b) : Method spesifik untuk menonton yang diturunkan dari induk dengan parameter tambahan.

void sukaMenonton(String a) : Method spesifik untuk menonton yang sama dengan induk.

void sukaMembaca(String a) : Method umum untuk membaca yang sama dengan induk.

4.2. Ubahlah Contoh 4 dengan menambahkan objek anak dengan method yang berbeda!

Untuk menambahkan objek anak dengan method yang berbeda, kita dapat menambahkan method lain di kelas anak yang tidak ada di kelas induk. Misalnya, kita dapat menambahkan method untuk suka bermain.

[No.4] Analisis dan Argumentasi

- Saya mengusulkan permasalahan ini dapat diatasi dengan cara Menambahkan method baru di kelas anak untuk menambahkan perilaku yang berbeda.
- Alasan solusi ini karena dengan menambahkan method baru di kelas anak, kita dapat menambahkan perilaku yang spesifik untuk anak tanpa mengubah perilaku induk.
- Perbaiki kode program dengan cara Membuat method baru di kelas anak dan memanggilnya di dalam main method.

[No.4] Penyusunan Algoritma dan Kode Program

- Algoritma
 - Deklarasikan kelas induk Ortu.
 - Tambahkan method spesifik dan umum di kelas induk.
 - Buat kelas anak Anak yang mengextends kelas induk.
 - Tambahkan method baru di kelas anak.
 - Dalam main method, buat objek dari kelas induk dan kelas anak, lalu panggil method yang relevan.
- Kode program dan luaran

```
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
10 public static void main(String[] args) {
11     System.out.println("Sifat Orang Tua :");
12     Ortu objekO = new Ortu(); // memanggil objek induk
13     objekO.sukaMenonton("Berita"); // memanggil sifat spesifik induk
14     objekO.sukaMembaca("Koran"); // memanggil method dengan variabel dapat diubah
15
16     System.out.println("\n Sifat Anak :");
17     Anak objekA = new Anak(); //memanggil objek anak
18     objekA.sukaMenonton(9, "Film Drakor"); //memanggil sifat spesifik anak yang diturunkan induk
19     objekA.sukaMembaca("Komik One Piece"); //memanggil method ke induk yang otomatis diturunkan tanpa deklarasi ulang di anak
20
21     // Tambahkan method baru di kelas anak
22     objekA.sukaBermain("Mainan Lego");
23 }
24
25 class Anak extends Ortu {
26     void sukaMenonton(int a, String b) {
27         System.out.println("Nonton Jam " + a + " Malam " + b);
28     }
29     void sukaMenonton(String a) { // method induk spesifik
30         System.out.println("Nonton " + a);
31     }
32     void sukaMembaca(String a) { // method induk umum bisa diubah anak
33         System.out.println("Suka Baca " + a);
34     }
35
36     // Tambahkan method baru di kelas anak
37     void sukaBermain(String mainan) {
38         System.out.println("Suka Bermain " + mainan);
39     }
40 }
41
42 public static void main(String[] args) {
43     System.out.println("Sifat Orang Tua :");
44     Ortu objekO = new Ortu(); // memanggil objek induk
45     objekO.sukaMenonton("Berita"); // memanggil sifat spesifik induk
46     objekO.sukaMembaca("Koran"); // memanggil method dengan variabel dapat diubah
47
48     System.out.println("\n Sifat Anak :");
49     Anak objekA = new Anak(); //memanggil objek anak
50     objekA.sukaMenonton(9, "Film Drakor"); //memanggil sifat spesifik anak yang diturunkan induk
51     objekA.sukaMembaca("Komik One Piece"); //memanggil method ke induk yang otomatis diturunkan tanpa deklarasi ulang di anak
52
53     // Panggil method baru di kelas anak
54     objekA.sukaBermain("Mainan Lego");
55 }
56 }
```

- Screenshot/ Capture potongan kode dan hasil luaran

Output Generated Files

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

Sifat Anak :
Nonton Jam 9 Malam Film Drakor
Suka Baca Komik One Piece
Suka Bermain Mainan Lego
```

Compiled and executed in 1.261 sec(s)

- b) Analisa luaran yang dihasilkan
Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data dan Program berhasil menampilkan informasi lengkap tentang objek Ortu Anak, serta perilaku yang berbeda.

[No.4] Kesimpulan

1) Analisa

- a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
- b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?
 - a) Kesimpulan berdasarkan permasalahan, algoritma, dan kode program menunjukkan bahwa penambahan method baru di kelas anak sangat penting untuk menambahkan perilaku yang spesifik tanpa mengubah perilaku induk.
 - b) Dasar alasan pengambilan keputusan saya untuk kasus ini adalah:
 - Pada program itu saya menggunakan bentuk kelas public karena diperlukan aksesibilitas dari luar kelas.
 - Perbaiki program dengan menambahkan method baru karena struktur Java mengharuskan penggunaan enkapsulasi dan polimorfisme untuk melindungi data dan menambahkan perilaku yang berbeda.

Refleksi

Pada minggu ini saya belajar banyak terkait pemrograman berorientasi objek pada bahasa pemrograman java. Tantangan utama ketika ingin membedakan konstruktor dan method, meski memiliki beberapa kesulitan, tetapi latihan ini membantu saya untuk menyelesaikan permasalahan tersebut. Secara keseluruhan pengalaman ini meningkatkan pemahaman.