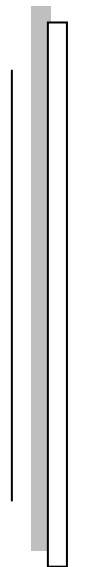




PEDOMAN PENYUSUNAN MODUL

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER (STMIK) SUMEDANG

DITETAPKAN DI SUMEDANG, 1 SEPTEMBER 2014 OLEH KETUA STMIK SUMEDANG



**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) SUMEDANG**

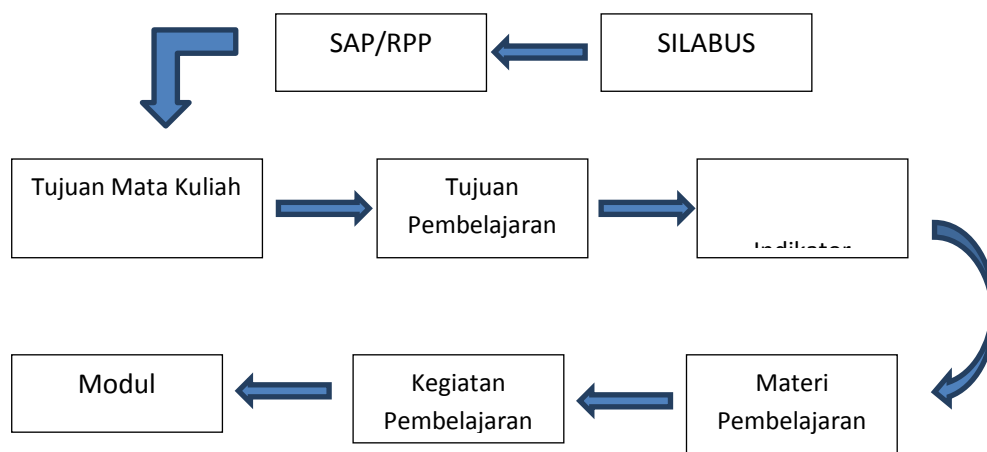
2014



A. PENGERTIAN

Modul merupakan alat atau sarana pembelajaran yang berisi materi, metode, batasan-batasan, dan cara mengevaluasi yang dirancang secara sistematis dan menarik untuk mencapai kompetensi yang diharapkan dan dapat digunakan secara mandiri. Modul disusun sesuai dengan kebutuhan belajar pada mata kuliah tertentu untuk keperluan proses pembelajaran tertentu, sebuah kompetensi atau sub kompetensi dikemas dalam satu modul secara utuh (self contained), mampu membelajarkan diri sendiri atau dapat digunakan untuk belajar secara mandiri (self instructional), penggunaannya tidak tergantung dengan media lain (self alone), memberikan kesempatan mahasiswa untuk berlatih dan memberikan rangkuman, memberi kesempatan melakukan tes sendiri (self test) dan mengakomodasi kesulitan mahasiswa dengan memberikan tindak lanjut dan umpan balik.

B. ALUR PENYUSUNAN



C. FORMAT MODUL

Bagian-bagian sebuah modul yang lengkap terdiri dari :

1. Sampul
2. Topik/materi belajar
3. Pengantar
4. Kompetensi dasar
5. Hasil akhir yang diharapkan
6. Kegiatan belajar
 - a. Uraian dan contoh
 - b. Latihan
 - c. Rangkuman
 - d. Tes formatif
 - e. Umpan balik dan tindak lanjut
7. Kunci jawaban
8. Daftar pustaka

Bagian-bagian tersebut di atas yang tidak boleh "dihilangkan" atau menjadi syarat minimal sebuah modul terdiri dari :

1. Kompetensi dasar
2. Hasil akhir yang diharapkan
3. Kegiatan belajar
 - a. Uraian dan contoh

- b. Latihan
 - c. Rangkuman
4. Daftar pustaka

D. PENGISIAN FORMAT

1. Sampul

- a. Lambang Sekolah Tinggi
- b. Modul mata kuliah apa?
- c. Kode mata kuliahnya apa?
- d. Modul ke berapa?
- e. Judul topik/materi belajarnya apa?
- f. Disusun oleh siapa?
- g. Nama sekolah tinggi
- h. Tahun penulisan

2. Judul Topik / Materi Belajar

Judul topik ditulis sesuai dengan materi belajar atau pokok bahasan dalam Rencana Pelaksanaan Pembelajaran (RPP) atau Satuan Acara Pembelajaran (SAP)

3. Pengantar

Pengantar berisi kedudukan modul dalam suatu mata kuliah, ruang lingkup materi modul, kaitan antara pokok bahasan satu dengan pokok bahasan lainnya, kaitan antara sub pokok bahasan satu dengan sub pokok bahasan lainnya, dan sebagainya

4. Kompetensi Dasar

Kompetensi dasar merupakan penjabaran standar kompetensi yang berupa tujuan yang hendak dicapai untuk 1 (satu) pertemuan atau lebih. Kompetensi dasar berbentuk pernyataan tentang maksud yang menggambarkan kemampuan tertentu pada mahasiswa setelah menyelesaikan pengalaman belajar tertentu. Kompetensi dasar dibuat dengan menggunakan kata kerja yang mengandung rumusan tingkah laku yang bersifat umum, mencakup materi yang luas dan dicapai dalam beberapa tahap. Kata kerja tingkah laku yang dimaksud, misalnya : Mengetahui, mengerti, memahami, menafsirkan, memperkirakan, menanggapi, meyakini, memamerkan dan sebagainya.

5. Hasil Akhir yang Diharapkan

Hasil akhir yang diharapkan adalah pernyataan tentang maksud yang menggambarkan kemampuan tertentu pada mahasiswa setelah menyelesaikan pengalaman belajar tertentu. Kemampuan tersebut disusun dengan format ABC (Audience, Behaviour, Content) dan menggunakan rumusan tingkah laku yang spesifik dan jelas. Rumusan tingkah laku harus jelas, artinya menggunakan kata kerja yang operasional, yaitu hanya menggunakan 1 (satu) kata kerja (tingkah laku) yang spesifik, dapat diamati dan dapat diukur.

Rumusan tingkah laku yang dimaksud dapat disajikan dalam berbagai aspek dan tingkatan, baik kognitif, afektif maupun psikomotorik, sebagai berikut : mempunyai tingkatan sebagai berikut :

- a. Tingkah laku yang berhubungan dengan aspek pengetahuan (cognitive) mempunyai tingkatan ingatan (C1): Mendefinisikan, mengidentifikasi, menggambarkan, menyatakan, menamakan; Pemahaman (C2): Meringkas, menyimpulkan, mengklasifikasi, menjelaskan, membedakan, membandingkan, menyontohkan; Penerapan (C3): Melaksanakan, menggunakan, menampilkan, menerapkan, menunjuk, mengerjakan; Analisis (C4): Menguraikan, menata, mengintegrasikan, memisahkan, merinci, menyusun, mengubah, menghubungkan ; Evaluasi (C6) : Memeriksa, mengkritik, menilai, menguji, mendeteksi, memantau, menyimpulkan; Mencipta (C6): Merancang, membangun, merancang, merencanakan, memproduksi, menemukan, membuat

- b. Tingkah laku yang berhubungan dengan aspek sikap (affective) mempunyai tingkatan sebagai berikut: Penerimaan(A1): Menyatakan, memilih, mengikuti; Tanggapan(A2): Menjawab, membahas, mengerjakan; Keyakinan(A3): Menguraikan, membedakan, mengajak; Berkarya(A4): Mengubah, mengatur, mengintegrasikan; Ketekunan(A5): Mengusulkan, mempraktekan, membuktikan.
- c. Tingkah laku yang berhubungan dengan aspek perbuatan (psychomotoric) mempunyai tingkatan sebagai berikut: Pemilihan(P1): Memilih, menggambarkan, memisahkan; Beraksi(P2): Melakukan, Menggerakkan, memperlihatkan; Pamer(P3): Memamerkan, memasang, mendiskusikan; Membangun(P4): Membangun, mengatur, menyusun; Mengatur(P5): Mengatur, mengukur, melukiskan, dan sebagainya

6. Kegiatan belajar

Setiap modul berisi beberapa sub kegiatan belajar sesuai dengan apa yang tercantum dalam SAP dengan ketentuan sebagai berikut :

- a. Sub topik : Ditulis singkat dan padat sesuai dengan kegiatan belajar yang tercantum dalam SAP.
- b. Uraian dan contoh : Di bawah uraian dan contoh ini diberikan judul sub-sub topik atau pokok bahasan yang merupakan bagian dari sub kegiatan belajar. Setiap sub topik atau pokok bahasan terdiri dari beberapa sub sub topik atau pokok bahasan.
- c. Setelah penulisan judul sub-sub topik atau pokok bahasan diberikan uraian yang disertai ilustrasi atau contoh-contoh aktual. Uraian diberikan dengan gaya bahasa sederhana dan komunikatif dalam bentuk bertutur sehingga penulis seolah-olah hadir di depan pembaca dan tengah menjelaskan materi pembelajarannya.
- d. Contoh-contoh harus disertakan dalam uraian sehingga memperkuat penguasaan terhadap materi pembelajaran yang disajikan. Penyajian contoh-contoh dapat dilakukan dengan gambar-gambar yang berkaitan.
- e. Latihan : Untuk latihan dapat diberikan butir-butir pertanyaan berbentuk esei sebagai latihan untuk menguasai materi pembelajaran yang disajikan dalam uraian dan contoh. Latihan harus disertai dengan petunjuk cara menyelesaikan butir-butir pernyataan dalam latihan tersebut.
- f. Rangkuman : Rangkuman berisi tentang pokok-pokok materi pembelajaran yang disajikan dalam uraian dan contoh
- g. Tes formatif : Tes formatif diberikan dalam rangka mengukur kemajuan hasil belajar yang dicapai dalam 1 (satu) unit pembelajaran. Butir-butir pertanyaan yang diberikan dalam rangka tes formatif berbentuk tes obyektif (Benar Salah, pilihan ganda, menjodohkan, melengkapi) sehingga memudahkan proses pengukuran (memberikan nilai). Tes yang baik mempunyai berbagai jenjang kesulitan, banyak menggunakan pertanyaan tingkat tinggi (sekitar 75%), disusun dari pertanyaan yang mudah ke sukar dan jelas bahasanya.
- h. Umpan balik dan tindak lanjut : Umpan balik dan tindak lanjut diberikan dalam bentuk rumus yang dapat digunakan untuk memaknai pencapaian hasil belajar sehingga dapat diberikan umpan balik dan tindak lanjut yang dapat dilakukan oleh pembaca.

7. Kunci jawaban

Kunci jawaban butir-butir pertanyaan yang diberikan dalam tes formatif diberikan agar mahasiswa dapat melakukan proses pengukuran oleh diri sendiri.

8. Daftar pustaka

Mencantumkan daftar pustaka yang digunakan sebagai sumber belajar dengan menuliskan nama lengkap pengarang buku, judul buku, kota penerbitan buku, nama penerbit buku, tahun penerbitan buku dan halaman sumber belajar tersebut diacu. Cara menuliskan daftar pustaka : nama lengkap pengarang buku (ditulis tanpa gelar, nama akhir menjadi nama pertama dan dibatasi koma), koma, judul buku (digaris bawah atau dicetak miring), koma, kurung buka, kota penerbitan buku, titik dua, nama penerbit buku, koma, tahun penerbitan buku, kurung tutup, koma, halaman (tuliskan halaman dan nomor halaman sumber belajar tersebut diacu). Contoh : Wiharto, Mulyo, Filsafat Ilmu dan Logika, (Jakarta : Esa-Press, 2008), halaman 6.

E. KETENTUAN LAIN

1. Setiap mata kuliah terdiri dari 14 (empatbelas) topik, sesuai dengan jumlah pertemuan dalam 1 (satu) semester. Sebuah topik dalam sebuah modul memuat beberapa sub topik. Modul gabungan harus mempunyai bobot materi yang sesuai dengan beberapa modul.
2. Modul ditulis menggunakan file microsoft office words (doc) versi 2003, huruf Arial ukuran 11 dan jarak antar baris (leading) 2 spasi. Modul dapat disertai dengan gambar (visual), suara (audio) atau video.
3. Untuk memperlancar proses pengunduhan (download) bahan pembelajaran dari internet ukuran modul berkisar 1 megabytes. Sebuah modul yang besar akan dipecah dalam file kecil sehingga maksimal berukuran 1 megabytes.
4. Modul diserahkan dalam bentuk softcopy kepada Ketua Jurusan untuk dinilai kelayakannya.

Ditetapkan : di Sumedang
Pada tanggal : 1 September 2014

Dwi Yuniarto, S.Sos., M.Kom

CONTOH MODUL



PEMROGRAMAN BERORIENTASI OBJEK

IF1042

MODUL 1

KELAS & OBJEK

DISUSUN OLEH : ESA FIRMANSYAH, S.T., M.KOM

STMIK SUMEDANG

2013

JUDUL TOPIK/ MATERI BELAJAR

1. KELAS DAN OBJEK
2. PEWARISAN
3. POLYMORFISME
4. PENGKAPSULAN
5. INTERFACE
6. PEMAKETAN

Modul 1. Kelas dan Objek

A. Tujuan

Mahasiswa diharapkan dapat:

1. Membuat kelas dengan beberapa variabel/atribut dan method
2. Menginstantiasi kelas menjadi objek

B. Teori Singkat

Dalam paradigma pemrograman berorientasi objek dikenal kelas dan objek. Kelas merupakan blue print dari objek-objek yang akan dibuat. Analogi kelas dan objek seperti rancangan model rumah dan pembangunan rumah-rumah, adapun proses pembuatan objek dari kelas dikenal dengan instantiasi.

Sebagai contoh kita ambil kelas manusia. Kelas manusia mempunyai atribut : nama. Selain itu kelas manusia juga mempunyai method: tampilkanNama, kerja, makan. Kasus diatas diimplementasikan dalam bahasa Java sebagai berikut

```
/*  
Disimpan dalam file "manusia.java"  
*/  
public class manusia  
{  
    public String nama;  
    public manusia(String n)  

```

Kelas Objek

instantiasi

(2)

Adapun kode untuk menginstantiasi kelas manusia menjadi objek Andi yang mengimplementasikan method: tampilkanNama dan makan adalah sebagai berikut.

```
/*  
Disimpan dalam file "andi.java"  
*/  
  
class andi  
{  
    public static void main(String arg[])  

```

Hasil eksekusi terhadap class andi adalah sebagai berikut:

```
Nama= Andi  
Nyam... nyam... nyam...
```

Method Abstrak

Method abstract adalah method yang belum mempunyai implementasi. Pendefinisian method abstrak adalah dengan menambahkan keyword abstract, seperti contoh dibawah ini:

```
abstract void BerangkatKerja();
```

Pada method abstract ini tidak didefinisikan/implementasi method BerangkatKerja tersebut (misalkan apakah jalan kaki, naik angkot, naik motor, naik mobil atau dengan cara lainnya.

C. Praktikum

1. Buatlah kelas mahasiswa dengan atribut/variabel= nama dan npm dan mempunyai method: tampilkanNama, tampilkanNpm, belajar, olahRaga, makan, minum.
2. Buatlah 3 objek untuk menginstantiasi kelas mahasiswa dengan atribut nama dan npm Anda sendiri dan 2 teman Anda dan mengimplementasikan beberapa method yang telah Anda definisikan dalam kelas mahasiswa
3. Kompilasi file kelas mahasiswa dan objek-objek yang telah Anda buat, kemudian jalankan objek-objek yang telah Anda buat !

Modul 2. Pewarisan

A. Tujuan

Praktikan diharapkan dapat:

3. Memahami dan mengimplementasikan konsep pewarisan
4. Memahami dan mengimplementasikan this dan super sesuai kebutuhan
5. Memahami dan mengimplementasikan method overriding

B. Teori Singkat

Salah satu kelebihan pemrograman berorientasi objek adalah penggunaan ulang kode-kode yang telah dibuat. Pewarisan adalah salah satu cara untuk menggunakan kode-kode yang telah

dibuat sebelumnya.

Sebagai contoh kelas manusia diturunkan menjadi kelas: programmer, tentara.

```
/*
Disimpan dalam file "programmer.java"
*/

public class programmer extends manusia
{
    public programmer(String n)
    {
        super(n);
    }
    public void kerja()
    {
        System.out.println("Tak...Tak...Klik..");
    }
    public void bersantai()
    {
        System.out.println("Game over, You lose...");
    }
}
```

```
/*
Disimpan dalam file "tentara.java"
*/
```

```
class tentara extends manusia
{
    public String pangkat;
    public tentara(String n, String p)
    {
        super(n);
        this.pangkat= p;
    }
}
```

```

    public String tampilkanPangkat()
(4)
    {
    return pangkat;
    }
    public void kerja()
    {
    System.out.println("Dor... Dor... Dor..");
    }
    }

```

Keyword super digunakan untuk memanggil metodh yang ada pada baseclass, sedangkan this menunjukkan atau mereferensi pada objek terkini

Andi adalah seorang programmer keturunan manusia, setelah dia makan lalu kerja dan terakhir dia bersantai dengan komputernya sehingga implementasi dengan kode javanya.

```

/*
Disimpan dalam file "andi.java"
*/
class andi
{
    public static void main(String arg[])
    {
        programmer andi= new programmer("Andi");
        System.out.println("Nama= "+ andi.tampilkanNama());
        andi.makan();
        andi.kerja();
        andi.bersantai();
    }
}

```

Kalau dieksekusi class Andi akan menghasilkan

```

Nama= Andi
Nyam... nyam... nyam...
Tak.Tak..Klik..
Game over, You lose...

```

Setelah Objek andi memberitahu namanya, dia makan dengan metodh warisan dari kelas manusia, kemudian dia kerja dengan metodh khusus kelas programmer dan terakhir dia bersantai juga dengan metodh khusus kelas programmer. Kelas turunan akan mewariskan atribut-atribut dan metodh-metodh parentclassnya/baseclass, akan tetapi dia tidak mewarisi konstruktor-konstruktornya sehingga ketika andi makan maka dia makan dengan metodh dari parentclassnya (manusia).

Akan tetapi ketika dia kerja, dia kerja dengan metodh baru yang didefinisikan khusus pada kelas programmer ("Tak...Tak...Klik.. "bukan "Kerja....kerjaa..."), inilah yang disebut dengan metodh overriding.

Lainhalnya dengan Andi, Badu adalah seorang tentara berpangkat kopral keturunan manusia. Suatu saat dia ditanya atasannya nama dan pangkatnya, kemudian dia disuruh makan dan kemudian diperintah untuk kerja, maka kode javanya :

```

/*
Disimpan dalam file "badu.java"
*/
public class badu
{
    public static void main(String arg[])
    {
        tentara badu= new tentara("Badu","Kopral");
        System.out.println("Nama= "+ badu.tampilkanNama());
        System.out.println("Pangkat= "+ badu.tampilkanPangkat());
        badu.makan();
        badu.kerja();
    }
}

```

Dalam kasus badu, selain badu mewarisi atribut nama dari manusia, sebagai tentara dia mempunyai atribut pangkat. Jadi dalam pewarisan, kita bisa menambah atribut-atribut baru dan juga bisa menambahkan method-method baru, bahkan mengoverride method yang ada pada parentclassnya.

Method Final

Method final adalah method yang tidak bisa dioverride oleh subclassnya. Pendefinisian method final dengan cara menambahkan keyword final didepan definisi method tersebut, seperti pada contoh berikut

```

Public final void tidur();
System.out.println("Zzz..Zzzz..Zzzzzzz");

```

Dengan cara seperti ini maka kita mendefinisikan bahwa method tidur adalah dengan cara

```

System.out.println("Zzz..Zzzz..Zzzzzzz");

```

Baik dilakukan oleh kelasnya sendiri maupun oleh anak cucu kelasnya.

C. Praktikum

1. Buatlah turunan dari kelas mahasiswa yang telah Anda buat pada modul 1 menjadi kelas: Pecinta alam dan buatlah atribut dan method-methodnya masing-masing sebanyak 2 buah
2. Buatlah 2 objek untuk menginstantiasi kelas pecinta alam dengan atribut nama dan npm Anda sendiri dan 1 teman Anda dengan mengimplementasikan beberapa method yang telah Anda definisikan dalam kelas kelasnya.
3. Kompilasi file kelas pecinta alam dan objek-objek yang telah Anda buat, kemudian jalankan objek-objek yang telah Anda buat !

Modul 3. Polymorfisme

A. Tujuan

Praktikan diharapkan dapat:

1. Memahami dan mengimplementasikan polymorfisme

B. Teori Singkat

Salah satu pilar Pemrograman Berorientasi Objek adalah polimorfisme yaitu kemampuan beberapa objek bertipe sama bereaksi secara berbeda terhadap “pesan” yang sama.

Sebagai contoh kita tambah lagi turunan dari manusia yaitu kelas sopir. Kelas sopir diimplementasikan dalam java

```
/*
Disimpan dalam file “sopir.java”
*/
public class sopir extends manusia
{
    public sopir(String n)
    {
        super(n);
    }
    public void kerja()
    {
        System.out.println("Ngung... Ngung... Ngung...Ciiit..");
    }
}
```

Dedi adalah seorang sopir keturunan manusia, untuk menginstantiasi objek dedi ditunjukkan dalam kode berikut.

```
/*
Disimpan dalam file “dedi.java”
*/
public class dedi
{
    public static void main(String arg[])
    {
        sopir dedi= new sopir("Dedi");
        System.out.println("Nama= "+ dedi.tampilkanNama());
        dedi.makan();
        dedi.kerja();
    }
}
```

Kemudian Andi sang programmer, Badu sang tentara dan Dedi sang sopir diperintahkan untuk bekerja, apa reaksinya?

Untuk melihat reaksi masing-masing, perhatikan kode java berikut!

```
/*
Disimpan dalam file “pekerja.java”
*/
public class pekerja
{
    public static void main(String args[])
    {
        manusia[] profesi= new manusia[3];
        profesi[0]=new programmer("Andii");
    }
}
```

```
profesi[1]=new tentara("Badu","Kopral");
profesi[2]=new sopir("Dedi");
for (int i=0; i<3; i++)
{
    profesi[i].kerja();
}
}
```

Hasil eksekusi para pekerja adalah sebagai berikut:

```
Tak...Tak...Klik...
Dor... Dor... Dor...
Ngung... Ngung... Ngung...Ciiit..
```

Message sama yang dikirimkan ke objek berbeda akan menghasilkan hasil yang berbeda, inilah yang disebut polimorfisme.

C. Praktikum

1. Buatlah kelas Rohis yang merupakan kelas turunan dari mahasiswa dan buatlah method-methodnya.
2. Buatlah 1 objek untuk menginstantiasi kelas rohis dan mengimplementasikan beberapa method yang telah Anda definisikan dalam kelas rohis.
3. Panggilah objek yang telah Anda buat pada kelas pecinta alam dan rohis untuk bekerja sehingga tampil efek-efek polimorfisme

Modul 4. Pengkapsulan

A. Tujuan

Praktikan diharapkan dapat:

1. Memahami dan menggunakan dengan benar hak akses public, protected dan private
2. Memahami kegunaan pengaturan hak akses dalam pengembangan aplikasi

B. Teori Singkat

Pilar terakhir dari 3 pilar Pemrograman Berorientasi Objek adalah pengkapsulan, dimana pengembang software dapat menyembunyikan detail suatu objek.

Hak akses public memungkinkan semua kelas mengaksesnya, hak akses protected hanya diberikan kepada kelasnya sendiri dan turunannya, serta kelas-kelas dalam satu paket. sedangkan private hanya boleh diakses oleh kelasnya sendiri.

Perhatikan kelas manusia dibawah ini!

```
/*
```

Disimpan dalam file "manusia.java"

```
*/
class manusia
{
    public String nama;

    public manusia(String n)
    {
        this.nama = n;
    }
    public String tampilkanNama()
    {
        return nama;
    }
    public void makan()
    {
        System.out.println("Nyam... nyam... nyam...");
    }
    public void kerja()
    {
        System.out.println("Kerja... kerjaaa...");
    }
    private void bunuhDiri()
    {
        System.out.println("Dor...bruk...");
    }
}
```

Andi adalah objek bentukan dari kelas manusia

```
/*
Disimpan dalam file "andi.java"
*/
class andi
{
    public static void main(String arg[])
    {
        manusia andi= new manusia("Andi");
        System.out.println("Nama= "+ andi.tampilkanNama());
        andi.makan();
    }
}
```

Apa yang terjadi jika hak akses private diakses oleh kelas lain?

C. Praktikum

1. Kompilasi dan jalankan kelas manusia dan andi!
2. Ubah hak akses makan dari public menjadi protected dan ulangi praktikum nomor 1! Apa yang terjadi? Jelaskan!
3. Tambahkan pada kelas andi untuk memanggil method bunuh diri, ulangi praktikum no 1.

Apa yang terjadi? Jelaskan!

Modul 5. Interface

A. Tujuan

Praktikan diharapkan dapat:

1. Memahami konsep pewarisan ganda
2. Memahami konsep interface
3. Mengimplementasikan Interface pada Java untuk melakukan pewarisan ganda

C. Teori Singkat

Untuk membuat suatu kelas dapat kita turunkan dengan pewarisan field-field dan methodh pada base classnya. Bagaimana kita membuat kelas yang menurunkan sifat dari beberapa base class? misalkan kita akan membuat kelas superman yang dia bisa membuat program layaknya programmer, dia juga ahli menggunakan senjata layaknya tentara, bahkan dia bisa terbang seperti elang (keturunan binatang)? Caranya adalah dengan pewarisan ganda. Dalam Java tidak dikenal pewarisan ganda, sehingga digunakan interface.

Contoh pewarisan ganda yang tidak benar dalam Java
class superman extends programmer, tentara, burung
{
}

Perhatikan contoh berikut:

```
/*  
Disimpan dalam file "superman.java"  
*/  
interface programmer {  
    void memrogram();  
}  
interface tentara {  

```

```
class superman implements programmer,tentara,burung {  
    public void memrogram(){};  
    public void menembak(){};  
    public void terbang(){};  
    public void buangKotoran(){};  
}
```

Kemudian kelas superman diinstantiasi menjadi objek bernama bejo, contoh kode program javanya sebagai berikut:

```
/*  
Disimpan dalam file "bejo.java"
```

```

*/
public class bejo
{
    public static void main(String arg[])
    {
        superman bj= new superman();
        bj.memrogram();
    {
        System.out.println("Implementasi memrogram ...tak..tik");
    }
    bj.menembak();
    {
        System.out.println("Implementasi menembak ...dor..dor");
    }
    bj.terbang();
    {
        System.out.println("Implementasi terbang.....Zap....");
    }
    }
}

```

Dalam kode diatas bejo menentukan sendiri cara mengimplementasi beberapa method dari interface yang telah didefinisikan dalam kelas superman (misalkan cara memrogramnya bagaimana, cara menembaknya bagaimana dan cara terbangnya seperti apa), selain itu bejo tidak berminat untuk mengimplementasikan method buangKotorang dari interface burung (misalnya karena burung biasa membuang kotoran di sembarang tempat).

Jadi interface dapat dianalogikan seperti menandatangani kontrak kerja, misalnya sebagai dosen dia wajib mengajar, membuat soal ujian dsb, akan tetapi cara mengajarnya dan membuat soalnya dilakukan terserah masing-masing dosen (tidak ditentukan dalam kontrak kerja)

C. Praktikum

Buatlah kelas mahasiswaSuper dengan menggunakan interface mahasiswa, atlit, wiraswasta, dan kemudian instantiasi menjadi objek dengan nama sesuai nama Anda

Modul 6. Pemaketan

A. Tujuan

Praktikan diharapkan dapat:

1. Memahami dan mengimplementasikan pemaketan
2. Memahami dan menggunakan kelas yang terdapat dalam paket tertentu

B. Teori Singkat

Ketika kelas-kelas yang dibuat semakin banyak dan semakin banyak, hal ini akan membuat struktur program menjadi rumit kalau tidak dikelola dengan baik. Untuk itu kelas-kelas disimpan dalam paket-paket tertentu, misalkan kelas programmerC, programmerJava, programmerPHP berada/disimpan dalam

paket programmer. Sementara marinir, kopasus, paskhas berada dalam paket tentara. Keuntungan pengaturan program dalam nama paket-paket adalah:

1. Terhindar dari konflik penamaan. Misalkan saja dalam membuat program kita menggunakan atau mengimport kelas dari luar yang dibuat oleh programmer lain, sehingga mungkin saja dalam penamaan kelas terjadi persamaan. Dengan menunjukkan nama lengkap paket/kelasnya maka tidak akan terjadi konflik penamaan
2. Teratur. Misalkan paket dosen terdapat kelas dosenPBO, dosenKalkulus, dosenEtika. Paket tentara terdapat marinir, kopasus, paskhas. Dengan struktur demikian, maka akan mempermudah ketika ketika kita akan menggunakan/mengimport kelas, misalkan saja kita akan mengimport kelas kopasus, tentunya kita mencari dalam paket tentara-bukan paket dosen.

Sebagai contoh kita akan membuat 2 paket: paket programmer dan paket tentara. Paket programmer terdiri dari kelas programmerC dan programmerJava.

```
/*
Disimpan dalam "programmer/programmerC.java"
*/
package programmer;
public class programmerC
{
    public programmerC()
    {
    }
    public void kerja()
    {
        System.out.println("Implementasi methodh kerja Programmer C ..");
    }
    public void bersantai()
    {
        System.out.println("Implementasi methodh bersantai Programmer .. ");
    }
}
/*
Disimpan dalam file "programmer/programmerJava.java"
*/
package programmer;
)
public class programmerJava
{
    public programmerJava()
    {
    }
    public void kerja()
    {
        System.out.println("Implementasi methodh kerja Programmer Java ..");
    }
    public void bersantai()
    {
        System.out.println("Implementasi methodh bersantai Programmer Java.. ");
    }
}
}
```

Sedangkan paket tentara terdapat kelas AngkatanDarat dan AngkatanLaut.

```

/*
Disimpan dalam file "tentara/kopasus.java"
*/
package tentara;
public class kopasus
{
public kopasus()
{
}
public void kerja()
{
System.out.println("Implementasi metodh kerja kopasus ....");
}
public void bersantai()
{
System.out.println("Implementasi metodh bersantai kopasus");
}
}

```

Kalau dieksekusi class Andi akan menghasilkan

```

/*
Disimpan dalam file "tentara/marinir.java"
*/
package tentara;
public class marinir
{
public marinir()
{
}
public void kerja()
{
) System.out.println("Implementasi metodh kerja marinir ..");
}
public void bersantai()
{
System.out.println("Implementasi metodh bersantai marinir ..");
}
}

```

Mengimport kelas

Untuk dapat mengimport kelas digunakan keyword import [nama paketnya]. Sebagai contoh instantiasi kelas programmerJava dalam paket programmer menjadi objek ahmed.

```

/*
Disimpan dalam file "ahmed.java"
*/
import programmer.programmerC;
class ahmed
{
public static void main(String arg[])
{
programmerC ahmed= new programmerC();
ahmed.kerja();
}
}

```

sehingga hasil eksekusinya adalah;

Implementasi metodh kerja Programmer C

Banyak sekali kelas-kelas yang telah dibuat oleh Sun yang dapat kita gunakan, misalkan untuk membuat windows bisa digunakan/import paket awt dan swing.

```
/*  
Disimpan dalam file "JavaOk.java"  
*/  
import javax.swing.*;  
public class JavaOk {  
    public static void main(String[] args) {  
        JFrame frame = new JFrame("Java?");  

```

yang akan menampilkan hasil sebagai berikut:

C. Praktikum

1. Buatlah paket mahasiswa yang isinya kelas IF, kelas MI, kelas SI dan buat metodh-metodhnya
2. Buatlah paket orangTua Siswa yang isinya kelas petani, nelayan, wiraswasta atau yang lainnya dan buat metodhnya.
3. Instantiasi sebuah objek dan beri nama dengan nama Anda yang mengimport dari paket mahasiswa
4. Instantiasi sebuah objek dan beri nama dengan nama Orang tua Anda yang mengimport dari paket orang Tua Siswa

Lampiran.

Konfigurasi PATH dan Penggunaan Java

Konfigurasi PATH dan CLASSPATH

Setelah Java Development Kit terinstall (misalkan di C:\JDK), konfigurasilah file autoexec.bat dengan menyeting PATH dan CLASSPATH nya.

```
SET PATH=%PATH%;C:\JDK\bin
```

```
SET CLASSPATH=.;"C:\JDK\lib"
```

Pada setting PATH dimaksudkan agar file-file executable di direktori "C:\JDK\bin" bisa dijalankan dari seluruh direktori kerja.

Pada setting CLASSPATH dimaksudkan agar class-class yang ada di "C:\JDK\lib" bisa diimport dari seluruh direktori kerja.

Perintah-Perintah Dasar

1. Mengkompile file java

```
Prompt:\javac [option] [source file]
```

Option boleh ada dan boleh juga tidak

Contoh: Untuk mengkompile file "tes.java"

```
Prompt:\javac tes.java
```

2. Menjalankan class (hasil kompilasi)

```
Prompt:\Java [-option] nama class
```

Option boleh ada dan boleh juga tidak

Contoh untuk menjalankan tes.class

```
Prompt:\Java tes
```

3. Baca manual/dokumentasi programnya untuk keterangan selengkapnya!