

LAPORAN JOBSHEET 12

MATKUL PRAKTIKUM PEMOGRAMAN WEB



DOSEN PEMBIMBING

Wilda Imama Sabilla, S.Kom., M.Kom.

NAMA MAHASISWA

Ahmad Dzul Fadhli Hannan

2341720106

KELAS TI-2E

POLITEKNIK NEGERI MALANG

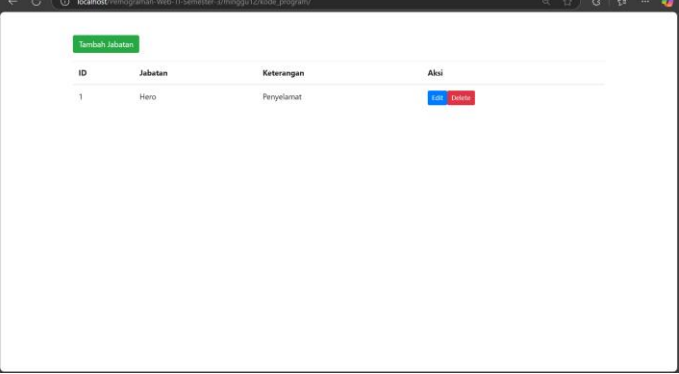
JURUSAN TEKNOLOGI INFORMASI

PRODI D4-TEKNIK INFORMATIKA

LINK GITHUB : https://github.com/adfhsjt/Pemograman-Web-TI-Semester-3/tree/main/minggu12/kode_program

No.	Jawaban
1.1	<p>Class car yang berisi atribut brand dan function startEngine yang Ketika dipanggil akan mencetak teks “Engine started!”, lalu terdapat objek car1 dan car2 yang merupakan instance dari class Car Dimana atribut brand car1 diset Toyota dan car2 Honda, lalu objek car1 memanggil fungsi startEngine dan car2 menampilkan atribut brand.</p>
1.2	<p>Class induk Animal yang berisi atribut protected name, lalu fungsi <code>__construct(\$name)</code> yang jika dipanggil akan set atribut name dengan parameter fungsi, dan fungsi eat dan sleep yang akan mencetak name Animal disertai eat atau sleep.</p> <p>Class anak Cat dan Dog yang merupakan extends dari Class Animal sehingga atribut dan fungsi dari class Animal juga bisa diakses lewat Class cat dan dog, dan didalam class cat terdapat fungsi meow dan dog bark yang menampilkan teks sesuai suara dari cat dan dog.</p> <p>Lalu dibuat objek cat name Whisker dan dog name Buddy, lalu dipanggil fungsi eat dan sleep dari class Animal. Lalu meow dan bark di class cat dan dog.</p>
1.3	<p>Interface Shape mendeklarasikan metode <code>calculateArea()</code>, yang wajib diimplementasikan oleh setiap kelas yang mengimplementasikan interface ini. Kelas Circle dan Rectangle masing-masing mengimplementasikan metode <code>calculateArea()</code> dengan cara yang sesuai dengan rumus luas masing-masing bentuk, yaitu πr^2 untuk lingkaran dan $\text{lebar} \times \text{tinggi}$ untuk persegi panjang. Fungsi <code>printArea()</code> menerima objek bertipe Shape sebagai parameter dan memanggil metode <code>calculateArea()</code> untuk mencetak hasil luas area dari objek tersebut, baik itu objek Circle maupun Rectangle.</p>
1.4	<p>kelas Car1 untuk mengelola atribut model dan warna mobil dengan cara yang aman dan terkontrol. Atribut <code>\$model</code> dan <code>\$color</code> dideklarasikan sebagai privat, sehingga hanya dapat diakses melalui metode khusus yang disediakan oleh kelas. Untuk mendapatkan nilai atribut tersebut, digunakan metode getter seperti <code>getModel()</code> dan <code>getColor()</code>, sementara untuk mengubah nilai atribut, digunakan metode setter seperti <code>setModel()</code> dan <code>setColor()</code>. Dengan cara ini, data terkait mobil terlindungi dari akses langsung dari luar kelas, yang memungkinkan kontrol lebih besar terhadap cara data diubah atau diakses. Ketika objek carNew dibuat dengan model "Toyota" dan warna "Blue", kita dapat melihat informasi tersebut melalui metode getter. Setelah itu, warna mobil diubah menjadi "Red" menggunakan metode setter, dan perubahan tersebut dapat segera terlihat dengan memanggil kembali metode <code>getColor()</code>.</p>
1.5	<p>Kelas abstrak Shape1 mendefinisikan sebuah metode abstrak <code>calculateArea()</code>, yang tidak memiliki implementasi tetapi harus diimplementasikan oleh setiap kelas yang meng-extend kelas ini. Kelas Circle1 dan Rectangle1 adalah dua contoh kelas yang meng-extend Shape1, masing-masing mengimplementasikan metode <code>calculateArea()</code> untuk menghitung luas lingkaran dan persegi panjang. Di dalam kelas Circle1, luas dihitung menggunakan rumus πr^2, sementara di kelas Rectangle1, luas dihitung dengan rumus $\text{lebar} \times \text{tinggi}$. Dengan menggunakan kelas abstrak, kita dapat memastikan bahwa setiap bentuk geometris memiliki metode <code>calculateArea()</code>, namun cara perhitungan luas disesuaikan dengan bentuk masing-masing.</p>
1.6	<p>Terdapat dua interface yang didefinisikan, yaitu Shape2 dan Color. Interface Shape2 memiliki metode <code>calculateArea()</code> yang wajib diimplementasikan oleh kelas yang mengimplementasikannya, sementara interface Color mendeklarasikan metode</p>

	getColor() untuk mendapatkan warna objek. Kelas Circle2 mengimplementasikan kedua interface ini, sehingga kelas ini harus menyediakan implementasi untuk kedua metode tersebut. Di dalam kelas Circle2, metode calculateArea() digunakan untuk menghitung luas lingkaran dengan rumus πr^2 , sementara metode getColor() mengembalikan warna lingkaran yang diset melalui konstruktor. Dengan cara ini, kelas Circle2 bisa menyediakan informasi baik tentang luas area maupun warna lingkaran. Objek circle2 dibuat dengan jari-jari 5 dan warna "Blue", kemudian kedua metode calculateArea() dan getColor() dipanggil untuk menampilkan informasi terkait objek tersebut.
1.7	Kelas Car2 memiliki atribut privat \$brand yang menyimpan merek mobil. Konstruktor __construct(\$brand) digunakan untuk menginisialisasi objek dengan merek mobil yang diberikan dan mencetak pesan "A new car is created." setiap kali objek baru dibuat. Metode getBrand() memungkinkan kita untuk mengakses merek mobil yang disimpan dalam atribut \$brand. Selain itu, kelas Car2 juga memiliki destruktur __destruct(), yang akan dipanggil secara otomatis saat objek dihancurkan atau keluar dari ruang lingkup, dan mencetak pesan "The car is destroyed." pada saat objek car2 dihancurkan setelah proses selesai. Dalam contoh ini, objek car2 dibuat dengan merek "Toyota", dan merek mobil tersebut ditampilkan menggunakan metode getBrand(). Setelah eksekusi selesai, destruktur secara otomatis akan dipanggil untuk menghancurkan objek dan mencetak pesan penghancuran objek.
1.8	Kelas Animal1 memiliki tiga atribut: \$name, \$age, dan \$color. Atribut \$name dideklarasikan sebagai public, yang artinya dapat diakses dari mana saja, baik dari dalam kelas maupun dari luar kelas. Atribut \$age dideklarasikan sebagai protected, sehingga hanya dapat diakses dari dalam kelas itu sendiri atau kelas turunan yang meng-extend kelas Animal1. Sedangkan atribut \$color dideklarasikan sebagai private, yang berarti hanya dapat diakses dari dalam kelas Animal1 saja dan tidak bisa diakses langsung dari luar kelas. Konstruktor __construct(\$name, \$age, \$color) digunakan untuk menginisialisasi ketiga atribut tersebut saat objek dibuat. Selain itu, terdapat metode getter getName(), getAge(), dan getColor() yang digunakan untuk mengakses nilai atribut-atribut tersebut, masing-masing sesuai dengan tingkat aksesibilitasnya. Objek animal1 dibuat dengan nama "Dog", usia 3, dan warna "Brown", kemudian informasi tentang objek ini ditampilkan dengan memanggil metode getter untuk masing-masing atribut.
	Output percobaan 1

	<p>1.1 Engine started!</p> <p>1.2 Honda Whiskers is eating. Buddy is sleeping. Whiskers says meow!. Buddy says woof!.</p> <p>1.3 Area: 78.539816339745 Area: 24</p> <p>1.4 Model: Toyota Color: Blue Updated Color: Red</p> <p>1.5 Area of Circle: 78.539816339745 Area of Rectangle: 24</p> <p>1.6 Area of Circle: 78.539816339745 Color of Circle: Blue</p> <p>1.7 A new car is created. Car brand: Toyota</p> <p>1.8 Name: Dog Age: 3 Color: Brown The car is destroyed.</p>
2.1	 <p>Jika klik edit</p>

Edit Jabatan

Jabatan

Keterangan

Perawatan

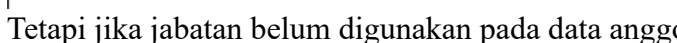
Update

1	Hero	Penyayang	Edit	Delete
---	------	-----------	----------------------	------------------------

prkwebdb.jabatan: 1 rows total (approximately)

[illegible]

Contoh error jika jabatan sudah digunakan pada data anggota



Dengan klik Tambah Jabatan.

The screenshot displays a web application interface. In the background, there is a table with two columns: 'ID' and 'Jabatan'. The first row contains the values '1' and 'Hero'. A green button labeled 'Tambah Jabatan' is visible in the top left corner. Overlaid on top of the table is a white modal window titled 'Tambah Data Jabatan' with a close button (X) in the top right corner. The modal contains two text input fields: 'Jabatan:' and 'Keterangan:'. At the bottom left of the modal is a blue button labeled 'Tambah!'. A red 'delete' button is partially visible on the right side of the table.

Maka akan muncul pop up form tambah jabatan.

Jabatan:

Knight

Keterangan:

Terobos Terus!!

Tambah

Dan jika klik tambah.

3	Knight	Terobos Terus!!
---	--------	-----------------

Tambah Jabatan

ID	Jabatan	Keterangan	Aksi
1	Hero	Penyayang	Edit Delete
3	Knight	Terobos Terus!!	Edit Delete

Data akan masuk di db dan ditampilkan di index.php. Dan data ini yang bisa dihapus karena tidak terkait dengan data lain dari anggota atau FK = null.

localhost/Pemrograman-Web-T1-Semester-3/minggu-12/Academic_program/index.php?action=delete

Tambah Jabatan

ID	Jabatan	Keterangan	Aksi
1	Hero	Penyayang	Edit Delete