

SOAL POSTEST
MK PEMOGRAMAN BERBASIS OBYEK
MODUL 12 – *Design Pattern*

Nama: Robin jauhari
Kelas: C
NPM: 231712550

GUIDED

1. Soal Singleton

Buatlah kelas Singleton untuk mencatat log aktivitas aplikasi. Hanya satu instance logger yang boleh ada dalam sistem.

Tugas:

Buat kelas Logger yang hanya memiliki satu instance. Tambahkan metode `log(message)` untuk mencatat pesan log. Pastikan log disimpan dalam sebuah array dan dapat diakses melalui metode `getLogs()`.

```
1  class Logger {
2      constructor() {
3          if (!Logger.instance) {
4              Logger.instance = this;
5              this.logs = [];
6          }
7          return Logger.instance;
8      }
9
10     log(message) {
11         // Tambahkan pesan ke daftar log
12     }
13
14     getLogs() {
15         // Kembalikan semua log
16     }
17 }
18
19 const logger1 = new Logger();
20 logger1.log("Aplikasi dimulai");
21 const logger2 = new Logger();
22 logger2.log("Pengguna login");
23
24 console.log(logger1 === logger2);
25 console.log(logger1.getLogs());
26 //SS outputnya
```

dan berikan Screenshoot untuk full kode beserta Outputnya

```
1 class Logger {
2   constructor() {
3     if (!Logger.instance) {
4       Logger.instance = this;
5       this.logs = [];
6     }
7     return Logger.instance;
8   }
9
10  log(message) {
11    const logs = new Logger(message);
12    this.logs.push(message);
13    return message;
14  }
15
16  getLogs() {
17    return this.logs;
18  }
19 }
20
21 const logger1 = new Logger();
22 logger1.log("Aplikasi dimulai");
23 const logger2 = new Logger();
24 logger2.log("Pengguna login");
25
26 console.log(logger1 === logger2);
27 console.log(logger1.getLogs());
28
```

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS

[Running] node "/root/pbo/gd12.js"

true

['Aplikasi dimulai', 'Pengguna login']

[Done] exited with code=0 in 0.186 seconds

2. Soal Observer

Implementasikan Pola Observer untuk sistem sensor cuaca, di mana stasiun cuaca memberi tahu perangkat yang terhubung setiap kali suhu berubah.

Tugas:

Buat kelas WeatherStation dengan metode addDevice, removeDevice, dan updateTemperature. Buat kelas Device yang menerima pembaruan suhu dari stasiun cuaca. Tambahkan beberapa perangkat dan perbarui suhu.

```

1  class WeatherStation {
2      constructor() {
3          this.devices = [];
4          this.temperature = 0;
5      }
6      addDevice(device) {
7          // Tambahkan perangkat ke daftar
8      }
9      removeDevice(device) {
10         // Hapus perangkat dari daftar
11     }
12     updateTemperature(temp) {
13         // Perbarui suhu dan beri tahu semua perangkat
14     }
15 }
16
17 class Device {
18     constructor(name) {
19         this.name = name;
20     }
21
22     notify(temp) {
23         console.log(`${this.name} menerima suhu baru: ${temp}`);
24     }
25 }
26
27 const station = new WeatherStation();
28 const phone = new Device("Phone");
29 const tablet = new Device("Tablet");
30
31 station.addDevice(phone);
32 station.addDevice(tablet);
33 station.updateTemperature(25);
34 station.removeDevice(phone);
35 station.updateTemperature(30);

```

dan berikan Screenshoot untuk full code beserta Outputnya

```

1  class WeatherStation {
2      constructor() {
3          this.devices = [];
4          this.temperature = 0;
5      }
6      addDevice(device) {
7          this.devices.push(device);
8      }
9      removeDevice(device) {
10         this.devices = this.devices.filter(obs => obs !== device);
11     }
12     updateTemperature(temp) {
13         this.temperature = temp;
14         this.devices.forEach(device => device.notify(temp));
15     }
16 }
17
18 class Device {
19     constructor(name) {
20         this.name = name;
21     }
22
23     notify(temp) {
24         console.log(`${this.name} menerima suhu baru: ${temp}`);
25     }
26 }
27
28 const station = new WeatherStation();
29 const phone = new Device("Phone");
30 const tablet = new Device("Tablet");
31
32 station.addDevice(phone);
33 station.addDevice(tablet);

```

```

[Done] exited with code=0 in 0.028 seconds

[Running] node "/root/coba1.js"
Phone menerima suhu baru: 25
Tablet menerima suhu baru: 25
Tablet menerima suhu baru: 30

[Done] exited with code=0 in 0.178 seconds

```

3. Soal Factory

Buat Pola Factory untuk menghasilkan objek pesanan makanan berdasarkan jenis makanan.

Tugas:

Buat kelas Food, Pizza, dan Burger. Buat kelas FoodFactory dengan metode createFood(type) untuk menghasilkan objek makanan. Tambahkan metode serve() untuk setiap objek makanan.

```
1 class Food {
2   constructor(name) {
3     this.name = name;
4   }
5
6   serve() {
7     // Cetak pesan umum untuk menyajikan makanan
8   }
9 }
10
11 class Pizza extends Food {
12   constructor() {
13     super("Pizza");
14   }
15
16   serve() {
17     console.log("Pizza disajikan dengan saus tomat.");
18   }
19 }
20
21 class Burger extends Food {
22   constructor() {
23     super("Burger");
24   }
25
26   serve() {
27     console.log("Burger disajikan dengan kentang goreng.");
28   }
29 }
30
31 class FoodFactory {
32   createFood(type) {
33     // Buat makanan berdasarkan jenis
34   }
35 }
36
37 const factory = new FoodFactory();
38 const pizza = factory.createFood("pizza");
39 const burger = factory.createFood("burger");
40
41 pizza.serve();
42 burger.serve();
```

Silahkan screenshot full kode beserta outputnya

UGD

1.

Soal: Sistem Manajemen Koneksi Database dengan Pola Singleton dan Factory

Sistem Manajemen Koneksi Database dengan Pola Singleton dan Factory. Anda diminta untuk membuat sistem manajemen koneksi database menggunakan kombinasi Pola Singleton dan Pola Factory. Berikut adalah persyaratannya:

1. Singleton:

- Hanya boleh ada satu instance dari ConnectionManager untuk mengelola semua koneksi database. ConnectionManager harus memiliki metode createConnection untuk membuat koneksi baru dan getConnections untuk mengambil semua koneksi yang ada.

2. Factory:

- Setiap koneksi database dibuat menggunakan sebuah Factory yang mendukung dua jenis database: MySQL dan PostgreSQL. Factory harus memiliki metode createDatabase(type) untuk membuat instance dari MySQL atau PostgreSQL.

3. Database:

- Setiap database (MySQL atau PostgreSQL) harus memiliki properti type, host, dan metode connect untuk mencetak pesan koneksi.

TOLONG DIPERHATIKAN PEMBUATAN OBJEK, ADANYA PERBEDAAN PEMBUATAN OBJEK AKAN DIANGGAP SALAH!!

```
const manager = new ConnectionManager();
manager.createConnection("MySQL", "localhost");
manager.createConnection("PostgreSQL", "127.0.0.1");

const connections = manager.getConnections();
connections.forEach(conn => conn.connect());

console.log(connections);
```

BERIKAN FULL KODE BESERTA PEMBUATAN OBJEKNYA

2.

Soal: Sistem Pengelolaan Pesanan dengan Pola Factory dan Observer

Anda diminta untuk membuat sistem pengelolaan pesanan restoran menggunakan kombinasi **Pola Factory** dan **Pola Observer**. Berikut adalah persyaratannya:

1. **Pola Factory:**

- Ada kelas `OrderFactory` untuk membuat objek pesanan.
- Sistem mendukung dua jenis pesanan: `DineIn` dan `TakeAway`.
- Pesanan memiliki properti `type`, `orderId`, dan `details` yang disesuaikan berdasarkan jenis pesanan.

2. **Pola Observer:**

- Sistem memiliki `OrderManager` sebagai subjek yang bertugas mengelola pesanan.
- `OrderManager` harus dapat menambahkan observer (misalnya, chef, kasir) yang menerima pembaruan setiap kali ada pesanan baru.
- Observer memiliki metode `notify(order)` untuk menerima informasi pesanan.

3. **Fungsi Sistem:**

- Pesanan dibuat menggunakan `OrderFactory` dan dikelola oleh `OrderManager`.
- Ketika pesanan baru dibuat, semua observer akan menerima notifikasi.

TOLONG DIPERHATIKAN PEMBUATAN OBJEK, ADANYA PERBEDAAN PEMBUATAN OBJEK AKAN DIANGGAP SALAH!!

```

const factory = new OrderFactory();
const manager = new OrderManager();

const chef = new Chef("Chef (NAMA KALIAN)\n");
const cashier = new Cashier("Cashier Lisa\n");

manager.addObserver(chef);
manager.addObserver(cashier);

const order1 = factory.createOrder("DineIn", "001", {
  table: 5,
  items: ["Pasta", "Salad"],
});
const order2 = factory.createOrder("TakeAway", "002", {
  customerName: "John",
  items: ["Burger"],
});

manager.createOrder(order1);
manager.createOrder(order2);

const orders = manager.getOrders();
console.log(orders);

```

BERIKAN FULL KODE BESERTA PEMBUATAN OBJEKNYA

Ketentuan Pengerjaan:

1. Pengumpulan maksimal Rabu, 26 Oktober 2024 pukul 23.59.
2. Yang dikumpulkan PDF GUIDED dan KODE JS DILETAKKAN DI 1 ZIP
3. Kumpulkan dalam format .zip dengan format penamaan file : 12_Y_XXXX.zip
 - a. Y = kelas Kalian
 - b. X = 4 Digit NPM Kalian
4. Ketentuan nama tidak sesuai (-5)
5. Screenshot tidak sesuai ketentuan soal (-5 / kesalahan).
6. Format/nama objek tidak sesuai ketentuan soal (-5/ kesalahan)
7. SELAIN ZIP akan dianggap tidak mengumpulkan
8. DAN JANGAN LUPA DIBERIKAN NAMA DAN NPM DI KODE JS NYA