



SOAL UJIAN TENGAH SEMESTER

MATA KULIAH: Pola Desain Perangkat Lunak	TANGGAL : Senin, 11 Oktober 2010
SEMESTER : Ganjil 2010/2011	WAKTU : 10:00 – 12:00 (120 menit)
SIFAT UJIAN : Open Book, Komp, Close USB	SOAL : Dibawa pulang sebagai tugas
DOSEN : Tjatur Kandaga (720080)	POINT : Setiap soal = 25 point

Selesaikanlah soal-soal berikut dengan membuat sebuah project di Desktop komputer anda (UTS_DP_nama_nrp) dengan masing2 soal dibuat sebagai package terpisah (no1, no2, no3, no4, no5). Anda boleh menggunakan Java atau C#.

1. Singleton Pattern

Sebuah Printer (berbentuk fisik) dapat *sharing* penggunaannya oleh beberapa aplikasi yang berbeda secara bersama-sama. Masing-masing aplikasi seolah-olah memiliki Printer-nya sendiri-sendiri yang berupa *virtualPrinter*, tetapi semua objek *virtualPrinter* yang dibuat sesungguhnya hanya mengacu pada satu objek Printer. Gunakanlah *Singleton Pattern* sehingga semua pembuatan objek *virtualPrinter* akan mengacu pada hanya 1 objek Printer. Method untuk mencetak ke printer tersebut adalah `cetakData(String data)`. Buat juga kelas test-nya yaitu `TestPrint`.

2. Decorator Pattern

Sebuah Restoran menyediakan menu nasi goreng yang dapat diberikan berbagai jenis tambahan sehingga menjadi berbagai menu nasi goreng seperti : nasi goreng ikan asin, nasi goreng petai, nasi goreng pedas, nasi goreng kentang, nasi goreng keju, termasuk berbagai kombinasinya seperti nasi goreng ikan asin pedas dll. Supaya menu dasar yaitu nasi goreng tidak diubah-ubah, dan supaya program kombinasi menu-nya baik, anda diminta menerapkan *Decorator Pattern* pada permasalahan ini. Anda perlu membuat kelas / kelas abstrak / *interface* : `INasiGoreng`, `NasiGoreng`, `NasiGorengDekorator`, `IkanAsinDekorator`, `PetaiDekorator`, dst., juga kelas test-nya yaitu `PesanNasiGoreng`.

3. Adapter Pattern

Pada `Client` terdapat data *array* dari objek `Contact` (dengan atribut `String:nama`, dan `String:noHp`), data tersebut akan diurutkan, tetapi *method* untuk mengurutkan data hanya dapat melakukan operasi pengurutan data *array String*. Supaya mereka dapat bekerja sama anda diminta untuk membuat solusinya dengan menggunakan *Adapter Pattern*. Method untuk mengurutkan data yaitu :

`urutNama(String[] nama, String[] noHp)`, dan `urutNoHp(String[] nama, String[] noHp)`.

Untuk membandingkan *String* anda dapat menggunakan *method* yang sudah disediakan pada *Java API*, yaitu `string.compareToIgnoreCase(String anotherString)`. Kelas `Client` sekaligus berfungsi sebagai kelas test.

4. Strategy Pattern

Algoritma untuk menghitung luas suatu bentuk merupakan bagian yang perlu dipisahkan dari bagian program lainnya. Gunakanlah *Strategy Pattern* untuk menentukan algoritma penghitungan luas mana yang akan dipakai pada saat *runtime*. Algoritma yang perlu disediakan untuk menghitung luas : segiempat, segitiga, dan lingkaran. Beri nama kelas/kelas abstrak/interface yang anda buat sbb: *Bentuk*, *SegiEmpat*, *SegiTiga*, *Lingkaran*, *HitungLuas*, *HLSegiEmpat*, *HLSegiTiga*, *HLLingkaran*, dan *CobaStrategyPattern* (sebagai kelas test).

Contoh kelas test-nya adalah sebagai berikut:

```
public class CobaStrategyPattern
{
    public static void main(String[] args)
    {
        int panjang=10, lebar=5, alas=6, tinggi=7, jari=10;

        SegiEmpat segi4 = new SegiEmpat(panjang, lebar);
        SegiTiga segi3 = new SegiTiga(alas, tinggi);
        Lingkaran lingk = new Lingkaran(jari);

        System.out.println("Luas segi4 : " + segi4.luas());
        System.out.println("Luas segi3 : " + segi3.luas());
        System.out.println("Luas lingk : " + lingk.luas());
    }
}
```

5. Observer Pattern

Dalam rangka mengembangkan diri Universitas Kristen Maranatha *go public* yaitu menawarkan sahamnya kepada masyarakat dengan cara melakukan *listing* di Bursa Efek Indonesia. Setiap terjadi perubahan nilai saham UKM, maka para pemegang saham akan diberikan notifikasi. Para pemegang saham terdiri dari berbagai kalangan diantaranya : Investor, Yayasan, Rektor, Dosen, dan Mahasiswa. Gunakan *Observer Pattern* untuk menyelesaikan masalah ini. Kelas/kelas abstrak/interface yang perlu anda buat : *Subject*, *SahamUKM* (sebagai *ConcreteSubject*), *Observer*, *Investor*, *Yayasan*, *Rektor*, *Dosen*, *Mahasiswa* dan *CobaObserverPattern* (sebagai kelas test).

Pada kelas *SahamUKM* terdapat *method* yang memantau perubahan nilai saham yaitu *perubahanNilaiSaham(String sifat, float nilai)*, dengan *sifat* dapat berisi "naik"/"turun", dan *nilai* merupakan nilai saham terbaru.

Untuk menyelesaikan kasus ini anda tidak diperbolehkan menggunakan *API* yang disediakan Java yaitu *java.util.Observer* dan *java.util.Observable*.

oOoOo Selamat mengerjakan, GBU OoOo