

# Soal Unit Testing Java TestNG

## Topik: Unit Testing menggunakan TestNG pada Aplikasi ATM Sederhana

### Deskripsi Umum:

Dalam soal ini, Anda diminta untuk membuat sebuah aplikasi ATM sederhana dalam bahasa pemrograman Java yang memiliki beberapa fungsionalitas dasar. Setelah aplikasi ATM selesai dibuat, Anda juga harus membuat unit test untuk menguji fungsionalitas dari aplikasi tersebut menggunakan framework TestNG.

### Instruksi Soal:

#### 1. Membuat Aplikasi ATM Sederhana:

- Buat sebuah kelas `ATM` yang memiliki atribut `saldo`.
- Kelas `ATM` harus memiliki konstruktor yang menerima satu parameter untuk menginisialisasi saldo awal.
- Implementasikan metode berikut dalam kelas `ATM`:
  - `double lihatSaldo()` : Mengembalikan saldo saat ini.
  - `void setorUang(double jumlah)` : Menambah saldo dengan jumlah yang diberikan jika jumlah tersebut valid (lebih besar dari 0). Jika jumlah tidak valid, lempar pengecualian `IllegalArgumentException`.
  - `void tarikUang(double jumlah)` : Mengurangi saldo dengan jumlah yang diberikan jika jumlah tersebut valid (lebih besar dari 0 dan tidak melebihi saldo saat ini). Jika jumlah tidak valid, lempar pengecualian `IllegalArgumentException`.

#### 2. Membuat Unit Test menggunakan TestNG:

- Buat kelas `ATMTest` yang akan menguji fungsionalitas dari kelas `ATM`.
- Gunakan anotasi `@BeforeMethod` untuk menginisialisasi objek `ATM` dengan saldo awal sebesar 1000 sebelum setiap metode pengujian dijalankan.
- Implementasikan metode pengujian berikut:
  - `testLihatSaldo()` : Menguji apakah metode `lihatSaldo()` mengembalikan saldo yang benar.

- `testSetorUang()` : Menguji apakah metode `setorUang()` berfungsi dengan benar ketika jumlah yang valid disetor.
- `testSetorUangNegatif()` : Menguji apakah metode `setorUang()` melempar pengecualian ketika jumlah yang negatif disetor.
- `testTarikUang()` : Menguji apakah metode `tarikUang()` berfungsi dengan benar ketika jumlah yang valid ditarik.
- `testTarikUangLebihDariSaldo()` : Menguji apakah metode `tarikUang()` melempar pengecualian ketika jumlah yang ditarik melebihi saldo saat ini.
- `testTarikUangNegatif()` : Menguji apakah metode `tarikUang()` melempar pengecualian ketika jumlah yang negatif ditarik.

### 3. Petunjuk Penilaian:

- Kode harus sesuai dengan standar penulisan kode yang baik dan dapat dibaca.
- Implementasi metode pada kelas `ATM` harus memenuhi deskripsi fungsionalitas yang diberikan.
- Unit test harus mencakup semua skenario pengujian yang dijelaskan.
- Gunakan framework TestNG untuk implementasi unit test.
- Pastikan setiap metode pengujian berjalan dengan sukses dan mengembalikan hasil yang diharapkan.

## Contoh Implementasi:

### ATM.java

```
public class ATM {
    private double saldo;

    public ATM(double saldoAwal) {
        this.saldo = saldoAwal;
    }

    public double lihatSaldo() {
        return saldo;
    }
}
```

```

    public void setorUang(double jumlah) {
        if (jumlah > 0) {
            saldo += jumlah;
        } else {
            throw new IllegalArgumentException("Jumlah harus lebih besar dari 0");
        }
    }

    public void tarikUang(double jumlah) {
        if (jumlah > 0 && jumlah <= saldo) {
            saldo -= jumlah;
        } else if (jumlah > saldo) {
            throw new IllegalArgumentException("Saldo tidak mencukupi");
        } else {
            throw new IllegalArgumentException("Jumlah harus lebih besar dari 0");
        }
    }
}

```

Selamat mengerjakan ujian! Pastikan Anda memahami setiap langkah dalam implementasi aplikasi ATM dan unit test menggunakan TestNG.