

LAPORAN
WEEK 2 PEMROGRAMAN BERBASIS OBJEK

Dibuat untuk memenuhi salah satu tugas mata kuliah Pemrograman Berbasis Objek yang
diampu oleh Bapak Zulkifli Arsyad

Oleh:

NAMA : Hafiz Zulhakim

NIM : 241511044

KELAS: 2B

PROGRAM STUDI : D3 TEKNIK INFORMATIKA

JURUSAN : TEKNIK KOMPUTER DAN INFORMATIKA



POLITEKNIK NEGERI BANDUNG

Tahun Ajaran 2025/2026

Main.java

```
import java.io.PrintStream;
```

```
public class Main {
```

```
    public Main() {
```

```
    }
```

```
    public static void main(String[] var0) {
```

```
       >NamaBarang var1 = new>NamaBarang("Proyektor", 4);
```

```
       >NamaBarang var2 = new>NamaBarang("Colokan", 5);
```

```
       >Peminjam var3 = new>Peminjam("Hakim", "241511044");
```

```
       >Petugas var4 = new>Petugas("Budi");
```

```
        var4.layaniPeminjaman(var3, var1, 1);
```

```
        var4.layaniPeminjaman(var3, var2, 1);
```

```
       >PrintStream var10000 = System.out;
```

```
       >String var10001 = var1.getNama();
```

```
        var10000.println("Sisa stok " + var10001 + ": " + var1.getStock());
```

```
        var10000 = System.out;
```

```
        var10001 = var2.getNama();
```

```
        var10000.println("Sisa stok " + var10001 + ": " + var2.getStock());
```

```
    }
```

```
}
```

NamaBarang.java

```
public class>NamaBarang{  
    private String nama;  
    private int stock;  
  
    //Construction  
    public>NamaBarang(String nama, int stock){  
        this.nama = nama;  
        this.stock = stock;  
    }  
  
    //Getter and Setter  
    public String getNama(){  
        return nama;  
    }  
  
    public void setNama(String nama){  
        this.nama = nama;  
    }  
  
    public int getStock(){  
        return stock;  
    }  
  
    public void setStock(int stock){  
        this.stock = stock;  
    }  
}
```

```
}

//Method mengurangi stock barang
public void kurangiStock(int jumlah){
    if (stock >= jumlah){
        stock -= jumlah;
    } else{
        System.out.println(nama + " sudah dipinjam semuanya");
    }
}
}
```

Peminjam.java

```
public class Peminjam{
    private String nama;
    private String NIM;

    //Constructor
    public Peminjam(String nama, String NIM){
        this.nama = nama;
        this.NIM = NIM;
    }

    //Getter and Setter
    public String getNama(){
```

```
        return nama;
    }
}
```

```
public void setNama(String nama){
    this.nama = nama;
}
```

```
public String getNIM(){
    return NIM;
}
```

```
public void setNIM(String NIM){
    this.NIM = NIM;
}
```

```
//Method
```

```
}
```

Petugas.java

```
public class Petugas {
    private String nama;
```

```
//Constructor
```

```
public Petugas(String nama){
    this.nama = nama;
```

```
}
```

```
//Getter
```

```
public String getNama(){
```

```
    return nama;
```

```
}
```

```
//Method
```

```
public void layaniPeminjaman(Peminjam peminjam,>NamaBarang barang, int jumlah) {
```

```
    barang.kurangiStock(jumlah);
```

```
    System.out.println("Petugas " + nama + " melayani " + peminjam.getNama() + "  
meminjam " + jumlah + " " + barang.getNama());
```

```
}
```

```
}
```

```
PS C:\Users\hakim\OneDrive - Politeknik Negeri Bandung\Collage\S3\Pemrograman Berorientasi Objek\Praktek\PB0codingan\Week_2\Tugas_Teori_Pemin  
Main  
Petugas Budi melayani Hakim meminjam 1 Proyektor  
Petugas Budi melayani Hakim meminjam 1 Colokan  
Sisa stok Proyektor: 3  
Sisa stok Colokan: 4
```

Lesson Learn

Dari pengerjaan tugas ini, dapat dipelajari bahwa penggunaan atribut private dan method public (getter/setter) memberikan kendali penuh terhadap bagaimana data dalam objek diakses dan dimodifikasi. Konsep enkapsulasi ini mencegah perubahan data secara sembarangan, misalnya stok barang hanya bisa berkurang melalui method khusus, bukan langsung diubah nilainya. Interaksi antar objek, seperti antara Petugas, Peminjam, dan NamaBarang, membentuk alur kerja yang jelas, di mana setiap class memiliki perannya masing-masing. Constructor memastikan setiap objek diinisialisasi dengan data awal yang valid, sehingga tidak ada objek yang dibuat dalam keadaan kosong. Pemisahan tanggung jawab antar class juga memberikan manfaat besar, karena Peminjam hanya fokus pada data mahasiswa yang meminjam, NamaBarang fokus pada detail barang yang bisa dipinjam, dan Petugas bertugas melayani transaksi peminjaman. Dengan struktur ini, kode menjadi lebih terorganisir, mudah dipahami, serta lebih mudah untuk dikembangkan di masa depan, sesuai dengan praktik pemrograman berorientasi objek yang baik.