

# **LAPORAN PRAKTIKUM PEMROGRAMAN BERBASIS OBJEK**

Diajukan untuk memenuhi salah satu tugas praktikum Mata Kuliah Pemrograman Berbasis Objek  
(Praktek)



**Disusun Oleh:**  
**Ibnu Hilmi Athaillah | 241511079**  
**Jurusan Teknik Komputer dan Informatika**

**Program Studi D-3 Teknik Informatika**  
**Politeknik Negeri Bandung**  
**2025**

## Daftar Isi

Daftar Isi .....	2
Link Source Code .....	3
Deskripsi dan Tujuan Aplikasi.....	4
Source Code Program .....	5
Output Program.....	9
Lesson Learn.....	10

## **Link Source Code**

Github : [https://github.com/ibung/W2\\_PBO\\_Praktek.git](https://github.com/ibung/W2_PBO_Praktek.git)

## Deskripsi dan Tujuan Aplikasi

Program ini dibuat untuk mendata barang-barang yang dipinjam oleh mahasiswa di Gedung JTK. Program terdiri dari tiga class:

- **Class Mahasiswa** → Menyimpan identitas mahasiswa berupa nama, NIM, dan prodi.
- **Class Peminjaman** → Menyimpan informasi barang yang dipinjam, seperti nama barang dan jumlah, serta mahasiswa yang meminjamnya.
- **Class Main** → Berfungsi sebagai eksekutor program. Pada class ini dibuat objek mahasiswa dan peminjaman, kemudian data ditampilkan ke layar.

### Tujuan Program:

- Membantu pencatatan barang yang dipinjam mahasiswa.
- Menerapkan konsep OOP dasar (class, object, constructor, getter, setter).
- Membiasakan penggunaan class Main sebagai titik masuk (entry point) program Java.

## Source Code Program

Nama File : mahasiswa.java

```
public class mahasiswa {
    private String nama;
    private long nim;
    private String prodi;

    // Constructor
    public mahasiswa(String nama, long nim, String prodi) {
        this.nama = nama;
        this.nim = nim;
        this.prodi = prodi;
    }

    // Getter
    public String getNama() {
        return nama;
    }

    public long getNim() {
        return nim;
    }

    public String getProdi() {
        return prodi;
    }

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

    public void setNim(long nim) {
        this.nim = nim;
    }

    public void setProdi(String prodi) {
        this.prodi = prodi;
    }

    // Method
    public void printProfil() {
        System.out.println("---- Mahasiswa ----" +
            "\nNama : " + nama +
            "\nNIM : " + nim +
            "\nProdi : " + prodi +
```

```
        "\n");  
    }  
}
```

Nama File : peminjaman.java

```
import java.time.LocalDate;  
import java.time.LocalTime;  
  
public class peminjaman {  
    private mahasiswa peminjam;  
    private String namaBarang;  
    private LocalDate tanggal;  
    private LocalTime jamPengambilan;  
    private LocalTime jamPengembalian;  
  
    // Constructor  
    public peminjaman(mahasiswa peminjam, String namaBarang, LocalDate  
tanggal, LocalTime jamPengambilan, LocalTime jamPengembalian) {  
        this.peminjam = peminjam;  
        this.namaBarang = namaBarang;  
        this.tanggal = tanggal;  
        this.jamPengambilan = jamPengambilan;  
        this.jamPengembalian = jamPengembalian;  
    }  
  
    // Getter  
    public mahasiswa getPeminjam() {  
        return peminjam;  
    }  
  
    public String getNamaBarang() {  
        return namaBarang;  
    }  
  
    public LocalDate getTanggal() {  
        return tanggal;  
    }  
  
    public LocalTime getJamPengambilan() {  
        return jamPengambilan;  
    }  
  
    public LocalTime getJamPengembalian() {  
        return jamPengembalian;  
    }  
}
```

```

// Setter
public void setPeminjam(mahasiswa peminjam) {
    this.peminjam = peminjam;
}

public void setNamaBarang(String namaBarang) {
    this.namaBarang = namaBarang;
}

public void setTanggal(LocalDate tanggal) {
    this.tanggal = tanggal;
}

public void setJamPengambilan(LocalTime jamPengambilan) {
    this.jamPengambilan = jamPengambilan;
}

public void setJamPengembalian(LocalTime jamPengembalian) {
    this.jamPengembalian = jamPengembalian;
}

// Method
public void printStruk() {
    System.out.println("---- Peminjaman ----" +
        "\nPeminjam      : " + peminjam.getNama() + " (" +
peminjam.getNim() + ")" +
        "\nProgram Studi   : " + peminjam.getProdi() +
        "\nNama Barang    : " + namaBarang +
        "\nTanggal       : " + tanggal +
        "\nJam Pengambilan : " + jamPengambilan +
        "\nJam Pengembalian : " + jamPengembalian);
}
}

```

Nama File : main.java

```

import java.time.LocalDate;
import java.time.LocalTime;

public class main {
    public static void main(String[] args) {
        // Data Mahasiswa
        mahasiswa m = new mahasiswa("Ibnu Hilmi", 241511079L, "D3
Informatika");
        m.printProfil();

        // Contoh peminjaman

```

```
        peminjaman p = new peminjaman(m, "Proyektor", LocalDate.of(2025, 8,
28), LocalTime.of(12, 0), LocalTime.of(13, 30));
        p.printStruk();
    }
}
```



## Output Program

```
---- Mahasiswa ----  
Nama   : Ibnu Hilmi  
NIM    : 241511079  
Prodi  : D3 Informatika  
  
---- Peminjaman ----  
Peminjam      : Ibnu Hilmi (241511079)  
Program Studi : D3 Informatika  
Nama Barang   : Proyektor  
Tanggal       : 2025-08-28  
Jam Pengambilan : 12:00  
Jam Pengembalian : 13:30
```

*Output 1*

```
---- Mahasiswa ----  
Nama   : Muhammad Hisyam Ramadhan  
NIM    : 241511666  
Prodi  : D3 Informatika  
  
---- Peminjaman ----  
Peminjam      : Muhammad Hisyam Ramadhan (241511666)  
Program Studi : D3 Informatika  
Nama Barang   : Kabel Terminal  
Tanggal       : 2025-08-28  
Jam Pengambilan : 12:00  
Jam Pengembalian : 13:30
```

*Output 2*

## Lesson Learn

Setelah membuat tiga class ini, beberapa hal yang aku pelajari adalah:

1. **Struktur Program Berbasis OOP**  
Program jadi lebih rapi dengan memisahkan data mahasiswa, data peminjaman, dan eksekusi utama ke dalam class berbeda.
2. **Fungsi Class Main**  
Aku baru belajar bahwa class Main adalah tempat mengeksekusi program, dan di sinilah objek-objek dari class lain dipakai.
3. **Membuat dan Menggunakan Objek**  
Dari class Main, aku bisa membuat objek mahasiswa lalu menghubungkannya ke objek peminjaman. Ini memberi gambaran bagaimana objek saling berinteraksi.
4. **Keamanan Data dengan Setter & Getter**  
Dengan private variabel ditambah setter & getter, data mahasiswa dan peminjaman lebih aman karena tidak bisa diubah sembarangan.
5. **Penerapan di Dunia Nyata**  
Program sederhana ini sebenarnya bisa dikembangkan menjadi sistem peminjaman berbasis database, jadi terasa bahwa OOP bukan sekadar teori tapi bisa dipakai di kasus nyata.