

Nama : Siti Fauziah Wulandari

NIM : 2309106038

Kelas : A2

Praktikum PBO

POSTTEST 4

Overloaded method beliObat (tanpa resep)

```
protected void beliObat(String namaPelanggan, int usia, String alamat, String noTelp, int idObat, int jumlah) {
    Obat obatDibeli = null;
    for (Obat obat : daftarObat) {
        if (obat.getIdObat() == idObat) {
            obatDibeli = obat;
            break;
        }
    }

    if (obatDibeli == null) {
        System.out.println(x:"Obat tidak ditemukan!");
        return;
    }

    if (jumlah > obatDibeli.getStok()) {
        System.out.println(x:"Stok tidak mencukupi.");
        return;
    }

    double totalHarga = jumlah * obatDibeli.getHarga();
    Transaksi transaksi = new Transaksi(
        daftarTransaksi.size() + 1, idPelanggan, idObat, jumlah, totalHarga, LocalDate.now(),
        statusTransaksi:"Selesai", metodePembayaran:"Transfer", statusPeng..."Selesai", alamat, namaPelanggan, usia, noTelp
    );

    daftarTransaksi.add(transaksi);
    obatDibeli.setStok(obatDibeli.getStok() - jumlah);
    System.out.println(x:"\nTransaksi berhasil dibuat!");
}
```

Overloaded method beliObat (dengan resep)

```
protected void beliObat(String namaPelanggan, int usia, String alamat, String noTelp, int idObat, int jumlah, String namaDokter, String namaKlinik) {
    Obat obatDibeli = null;
    for (Obat obat : daftarObat) {
        if (obat.getIdObat() == idObat) {
            obatDibeli = obat;
            break;
        }
    }

    if (obatDibeli == null) {
        System.out.println(x:"Obat tidak ditemukan!");
        return;
    }

    if (jumlah > obatDibeli.getStok()) {
        System.out.println(x:"Stok tidak mencukupi.");
        return;
    }

    LocalDate tanggalResep = LocalDate.now();
    ResepDokter resep = new ResepDokter(idResep++, idPelanggan, idObat, namaDokter, namaKlinik, tanggalResep, statusVerifikasi:false);
    daftarResep.add(resep);
    System.out.println(x:"Resep berhasil diunggah, menunggu verifikasi Admin.");

    double totalHarga = jumlah * obatDibeli.getHarga();
    Transaksi transaksi = new Transaksi(
        daftarTransaksi.size() + 1, idPelanggan, idObat, jumlah, totalHarga, LocalDate.now(),
        statusTransaksi:"Menunggu Verifikasi Resep", metodePembayaran:"Transfer", statusPeng..."Belum Diambil", alamat, namaPelanggan, usia, noTelp
    );

    daftarTransaksi.add(transaksi);
    obatDibeli.setStok(obatDibeli.getStok() - jumlah);
    System.out.println(x:"\nTransaksi berhasil dibuat!");
}
```

Overloading diterapkan dalam class PelangganService melalui method beliObat, yaitu satu versi dengan parameter untuk pembelian obat biasa dan satu versi dengan parameter tambahan (nama dokter dan klinik) untuk pembelian obat dengan resep.

Override di Method toString()

```
@Override
public String toString() {
    DecimalFormat rupiahFormat = new DecimalFormat(pattern:"#,###.00");
    return "Obat{" +
        "ID=" + idObat +
        ", Nama='" + namaObat + '\'' +
        ", Harga=Rp " + rupiahFormat.format(harga) +
        ", Stok=" + stok +
        ", Butuh Resep=" + (resepDokter ? "Ya" : "Tidak") +
        '}';
}

public void tampilkanObat() {
    System.out.println(this);
}
}
```

Override terjadi pada method toString() di class Obat, yang menimpa method toString() milik class Object. Ini dilakukan agar saat objek dicetak (misalnya menggunakan System.out.println(obat)), informasi yang muncul lebih detail dan mudah dipahami, bukan hanya nama class dan hashcode seperti bawaan default Jav