Nama : Ruchil Amelinda

NIM : 2211522006

Kelas : PBO-B

**Materi *Java Database Connectivity* (JDBC)**

*Java Database Connectivity* (JDBC) adalah Java API yang digunakan untuk menghubungkan dan mengeksekusi *query* dengan *database*. JDBC memudahkan untuk mingirim *statement* SQL ke sistem databse relasional dan mendukung bermacam macam dialek SQL. Berikut ini beberapa keuntungan penggunaan JDBC :

* Mempertahankan data enterprise yang telah ada
* Menyederhanakan development enterprise
* Tidak pelru konfigurasi pada jaringan komputer
* Akses penuh ke metadata
* Koneksi database menggunkan URL (*Uniform Resource Locator).*

Selain itu, terdapat 4 jenis kelas *drivers*  yang digunakan pada JDBC diantaranya:

* JDBC-ODBC Bridge Driver
* Native class
* Network Protocol Driver
* Thin Drive

**Contoh Program JDBC**

**Interface//pembayaran**

public interface Pembayaran {

    void hitungTotalBayar ();

    void detail ();

}

**Induk class//Barang**

public class Barang {

    public static String namaBarang;

    public static String idBarang;

    public static int hargaBarang;

    public int jumlahBarang;

    public Barang( String namaBarang, String idBarang, int hargaBarang) {

        this.namaBarang=namaBarang;

        this.idBarang=idBarang;

        this.hargaBarang=hargaBarang;

    }

    public String getNamabarang() {

        return namaBarang;

    }

    public String getIdBarang() {

        return idBarang;

    }

    public int getHargabarang() {

        return hargaBarang;

    }

    public int getJumlahBarang(){

        return jumlahBarang;

    }

    public double getTotalHarga() {

        return hargaBarang \* jumlahBarang;

    }

}

**Anak class//Kasir**

public class Barang {

    public static String namaBarang;

    public static String idBarang;

    public static int hargaBarang;

    public int jumlahBarang;

    public Barang( String namaBarang, String idBarang, int hargaBarang) {

        this.namaBarang=namaBarang;

        this.idBarang=idBarang;

        this.hargaBarang=hargaBarang;

    }

    public String getNamabarang() {

        return namaBarang;

    }

    public String getIdBarang() {

        return idBarang;

    }

    public int getHargabarang() {

        return hargaBarang;

    }

    public int getJumlahBarang(){

        return jumlahBarang;

    }

    public double getTotalHarga() {

        return hargaBarang \* jumlahBarang;

    }

}

**Driver class//Mini market**

import java.lang.reflect.Method;

import java.sql.Connection;

import java.sql.Date;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.text.SimpleDateFormat;

import java.util.HashMap;

import java.util.InputMismatchException;

import java.util.Map;

import java.util.Random;

import java.util.Scanner;

public class Minimarket {

    public String driver = "com.mysql.jdbc.Driver";

    public String url = "jdbc:mysql://localhost:3307/jdbc";

    public static String uname = "root";

    public static String pass = "";

    //Method untuk menambahkan barang ke database

    public void tambahBarang(String namaBarang, double hargaBarang){

        try (Connection connection = DriverManager.getConnection(url, uname, pass);

             PreparedStatement statement = connection.prepareStatement("INSERT INTO barang (nama, harga) VALUES (?, ?)")) {

            statement.setString(1, namaBarang);

            statement.setDouble(2, hargaBarang);

            int rowsInserted = statement.executeUpdate();

            if (rowsInserted > 0) {

                System.out.println("Barang berhasil ditambahkan!");

            }

        } catch (SQLException e) {

            e.printStackTrace();

        }

    }

    //Method untuk menampilkan daftar barang dari database

    public void lihatBarang() {

        try (Connection connection = DriverManager.getConnection(url, uname, pass);

             Statement statement = connection.createStatement();

             ResultSet resultSet = statement.executeQuery("SELECT \* FROM barang")) {

            while (resultSet.next()) {

                System.out.println("ID: " + resultSet.getInt("id\_barang") +

                        ", Nama: " + resultSet.getString("nama") +

                        ", Harga: " + resultSet.getDouble("harga"));

            }

        } catch (SQLException e) {

            e.printStackTrace();

        }

    }

    //Method untuk memperbaharui harga barang berdasarkan ID

    public void updateHargaBarang(int id, double hargaBaru) {

        try (Connection connection = DriverManager.getConnection(url, uname, pass);

             PreparedStatement statement = connection.prepareStatement("UPDATE barang SET harga = ? WHERE id\_barang = ?")) {

            statement.setDouble(1, hargaBaru);

            statement.setInt(2, id);

            int rowsUpdated = statement.executeUpdate();

            if (rowsUpdated > 0) {

                System.out.println("Harga barang berhasil diupdate!");

            }

        } catch (SQLException e) {

            e.printStackTrace();

        }

    }

    //Method untuk menghapus barang berdasarkan ID

    public void hapusBarang(int id) {

        try (Connection connection = DriverManager.getConnection(url, uname, pass);

             PreparedStatement statement = connection.prepareStatement("DELETE FROM barang WHERE id\_barang = ?")) {

            statement.setInt(1, id);

            int rowsDeleted = statement.executeUpdate();

            if (rowsDeleted > 0) {

                System.out.println("Barang berhasil dihapus!");

            }

        } catch (SQLException e) {

            e.printStackTrace();

        }

    }

    private static Map<String, String> userDatabase = new HashMap<>();

    private static final int CAPTCHA\_LENGTH = 6;

    /\*\*

     \* @param args

     \*/

    public static void main(String[] args) {

        userDatabase.put("karyawan1", "password11");

        userDatabase.put("karyawan2", "password22");

        Scanner scanner = new Scanner(System.in);

        boolean loggedIn = false;

        while (!loggedIn) {

            System.out.print("Username: ");

            String username = scanner.nextLine();

            System.out.print("Password: ");

            String password = scanner.nextLine();

            if (login(username, password)) {

                loggedIn = true;

                System.out.println("Login berhasil!");

                System.out.print("No. Faktur        : ");

                String noFaktur = scanner.nextLine();

                System.out.print("Nama Pelanggan    : ");

                String namaPelanggan = scanner.nextLine();

                System.out.print("No. HP            : ");

                String nomorHP = scanner.nextLine();

                System.out.print("Alamat            : ");

                String alamatPelanggan = scanner.nextLine();

                System.out.print("Kode Barang       : ");

                String kodeBarang = scanner.nextLine();

                System.out.print("Nama Barang       : ");

                String namaBarang = scanner.nextLine();

                double hargaBarang;

            while (true) {

                try {

                    System.out.print("Harga Barang      : ");

                    hargaBarang = scanner.nextDouble();

                    break;

                } catch (InputMismatchException e) {

                    System.out.println("Input harga tidak valid. Mohon masukkan angka.");

                    scanner.next();

                }

            }

            // Buat objek transaksi

            Kasir transaksi = new Kasir(noFaktur, namaPelanggan, alamatPelanggan, nomorHP, kodeBarang, namaBarang, hargaBarang);

            // Hitung total bayar

            transaksi.hitungTotalBayar();

            // Tampilkan detail transaksi

            transaksi.detail();

            // Tanggal dan Waktu

            Date date = new Date(0);

            SimpleDateFormat hari = new SimpleDateFormat("'Hari/Tanggal \t:' EEEEEEEEEE dd-MM-yyyy");

            SimpleDateFormat jam = new SimpleDateFormat("'Waktu \t\t:' hh:mm:ss z");

            // Tampilkan Struk

            System.out.println(" ");

            System.out.println(" ");

            System.out.println("============== MINI MARKET ===============");

            System.out.println(hari.format(date));

            System.out.println(jam.format(date));

            System.out.println("No Faktur \t: " + noFaktur);

            System.out.println("========================================");

            System.out.println(" ");

            System.out.println("  ");

            System.out.println("------------ DATA PELANGGAN ------------");

            System.out.println("Nama Pelanggan \t: " + namaPelanggan);

            System.out.println("No. HP \t\t: " + nomorHP);

            System.out.println("Alamat \t\t: " + alamatPelanggan);

            System.out.println(" ");

            System.out.println("-------- DATA PEMBELIAN BARANG ---------");

            System.out.println("Kode Barang \t: " + kodeBarang);

            System.out.println("Nama Barang \t: " + namaBarang);

            System.out.println("Harga \t\t: " + hargaBarang);

            System.out.println("Total Bayar \t: " + transaksi.getTotalHarga());

            System.out.println("========================================");

            System.out.println("Kasir \t\t: Budi \n");

            System.out.println("");

            System.out.println("\t\t TERIMA KASIH \t\t");

            System.out.println("");

            // Method string

            System.out.println("toUpperCase\t: " + namaPelanggan.toUpperCase());

        int choice;

        } else {

            System.out.println("Login gagal. Apakah Anda ingin mencoba login dengan username dan password baru? (y/n)");

            String response = scanner.nextLine().toLowerCase();

            if (!response.equals("y")) {

                System.out.println("Terima kasih. Selamat tinggal.");

                    break;

            }

                // Memasukkan username dan password baru

                System.out.print("Masukkan username baru: ");

                String newUsername = scanner.nextLine();

                System.out.print("Masukkan password baru: ");

                String newPassword = scanner.nextLine();

                // Menambahkan pengguna baru ke dalam database

                userDatabase.put(newUsername, newPassword);

                // Menampilkan captcha

                String captcha = generateCaptcha();

                System.out.println("Captcha: " + captcha);

                System.out.print("Masukkan captcha: ");

                String userCaptcha = scanner.nextLine();

                if (captcha.equals(userCaptcha)) {

                    System.out.println("Pengguna baru ditambahkan dengan sukses!");

                } else {

                    System.out.println("Verifikasi captcha gagal. Coba lagi nanti.");

                }

            }

        }

    }

    private static boolean login(String username, String password) {

        // Mengambil password yang disimpan untuk username tertentu

        String storedPassword = userDatabase.get(username);

        // Memeriksa apakah username ditemukan dan password cocok

        return storedPassword != null && storedPassword.equals(password);

    }

    private static String generateCaptcha() {

        // Generate a random captcha string

        StringBuilder captcha = new StringBuilder();

        Random random = new Random();

        for (int i = 0; i < CAPTCHA\_LENGTH; i++) {

            char randomChar = (char) (random.nextInt(26) + 'A'); // Generate a random uppercase letter

            captcha.append(randomChar);

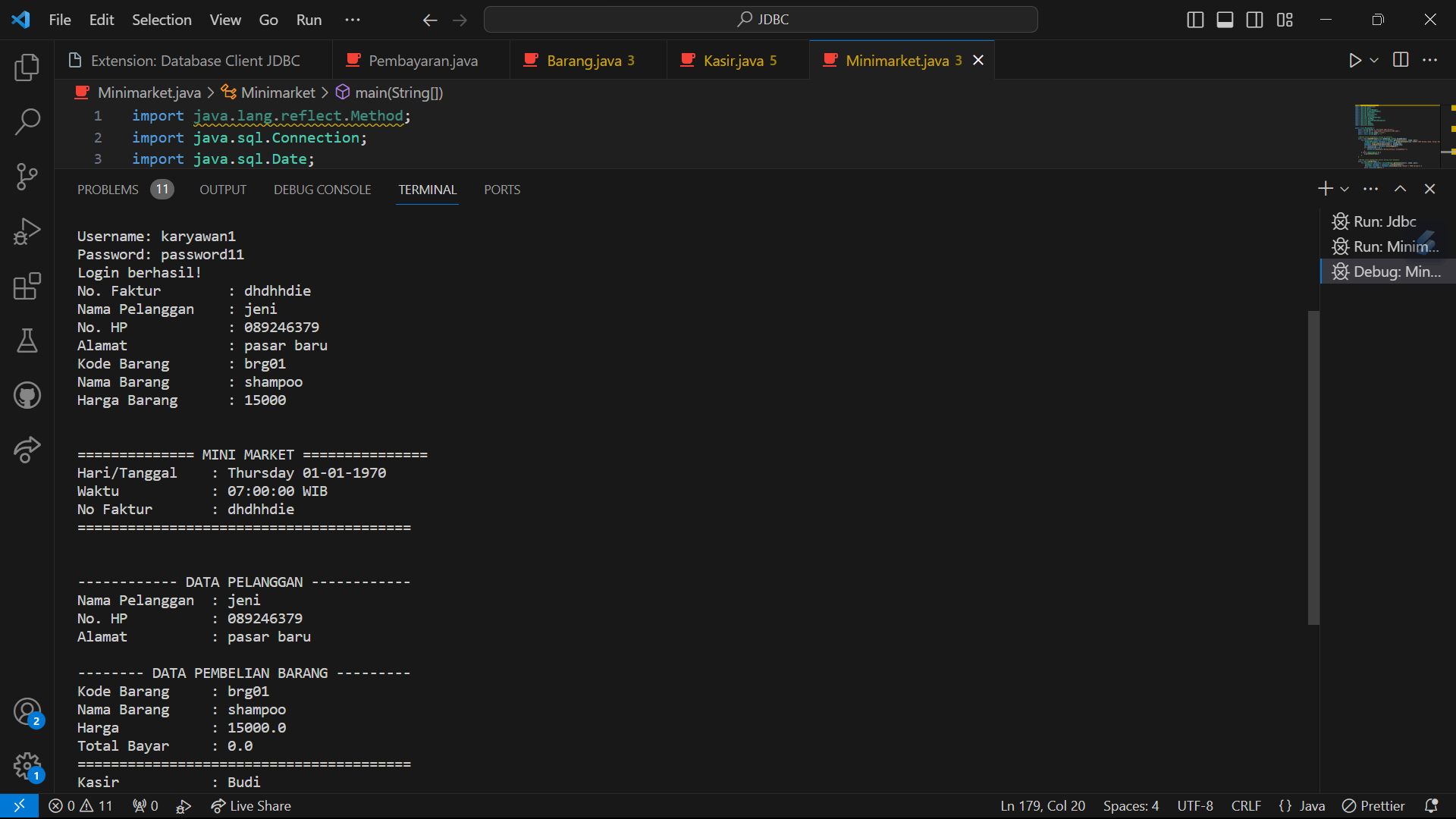
        }

        return captcha.toString();

    }

}

**Output**

****