

Tugas Pendahuluan Modul 15

PEMROGRAMAN BERORIENTASI OBJEK – Genap 2024/2025

“JDBC Part 3”

Nama : Muhammad Daffa Fariza

Kelas : IF-47-03

NIM : 103012300004

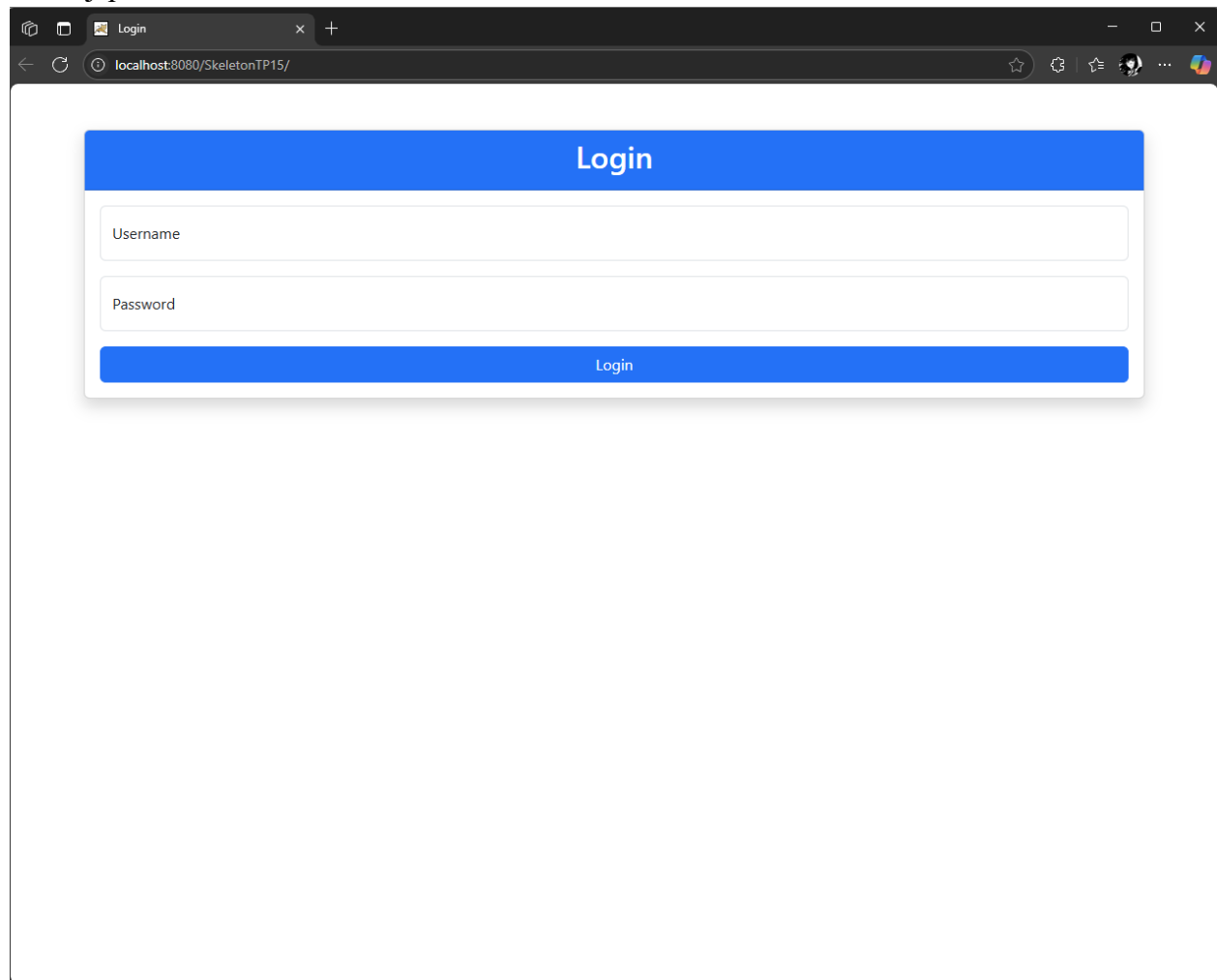
F. Uji Skenario

Lakukan uji coba akhir dengan scenario sebagai berikut:

1. Lakukan screenshot terhadap hasil akhir tampilan setiap tampilan (index, view, add, edit) anda.

Jawaban :

- Index.jsp



- View.jsp



Daftar Barang

Tambah Barang

Logout

ID	Nama	Harga	Aksi	
1	Laptop Dell	12500000	Edit	Hapus
2	Mouse Logitech	250000	Edit	Hapus
3	Keyboard Mechanical	850000	Edit	Hapus
4	Monitor Samsung 24"	1750000	Edit	Hapus
5	Flashdisk 64GB	120000	Edit	Hapus

- Edit.jsp

Browser window showing the "Edit Barang" page. The URL is `localhost:8080/SkeletonTP15/product?menu=edit&id=1`.

Edit Barang

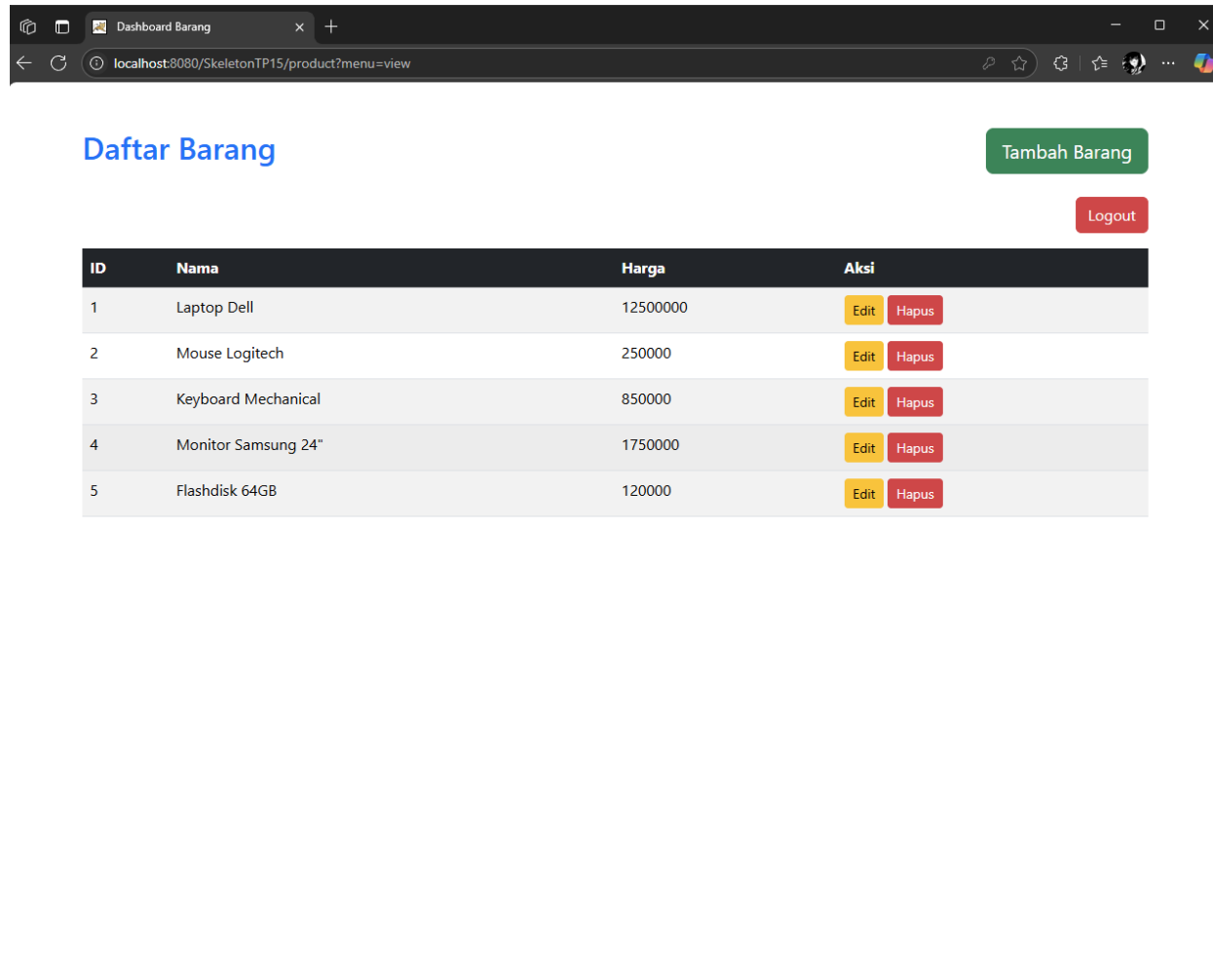
Nama Barang
Laptop Dell

Harga Barang
12500000

Kembali

Simpan Perubahan

- Add.jsp



2. Lakukan screenshot hanya pada kodingan setiap poin (#1.1) sampai (#3.9).

Jawaban :

1.1

```
29 public void connect() {  
30     try {  
31         Class.forName(className: "com.mysql.jdbc.Driver");  
32  
33         // (#1.1)  
34         // Isi nama database, user, dan password sesuai dengan  
35         // konfigurasi MySQL yang sudah terinstall pada perangkat masing-masing.  
36         con = DriverManager.getConnection( url:"jdbc:mysql://localhost:3306/db_barang", user: "root", password: "");  
37         stmt = con.createStatement();  
38         isConnected = true;  
39         message = "Database Terkoneksi";  
40     } catch (ClassNotFoundException | SQLException e) {  
41         isConnected = false;  
42         message = e.getMessage();  
43     }  
44 }
```

1.2

```

11  public Product() {
12      // (#1.2)
13      // Sesuaikan nilai table dan primaryKey
14      this.table = "product";
15      this.primaryKey = "id";
16  }

```

1.3

```

18  public Product(int id, String name, double price) {
19      // (#1.3) Sesuaikan nilai table dan primaryKey
20      // serta konstruktor dari parameter yang telah diberikan
21      this.table = "product";
22      this.primaryKey = "id";
23      this.id = id;
24      this.name = name;
25      this.price = price;
26  }

```

1.4

```

30  public Product toModel(ResultSet rs) {
31      try {
32          return new Product(
33              // (#1.4)
34              // Lakukan get resultSet dari tiap parameter kolom yang ada
35              id: rs.getInt(string: "id"),
36              name: rs.getString(string: "name"),
37              price: rs.getInt(string: "price")
38          );
39      } catch (SQLException e) {
40          System.out.println("Error: " + e.getMessage());
41          return null;
42      }

```

2.1

```

16  protected void doGet(HttpServletRequest request, HttpServletResponse response)
17      throws ServletException, IOException {
18      // (#2.1) Lakukan send redirect langsung ke jsp index
19      response.sendRedirect(location: "index.jsp");

```

2.2

2.3

2.4

2.5

2.6

2.7

2.8

```

23 protected void doPost(HttpServletRequest request, HttpServletResponse response)
24     throws ServletException, IOException {
25     String action = request.getParameter(name: "action");
26     // (#2.2) Lakukan pengecekan apakah var action bernilai "login"
27     if ("login".equals(anObject: action)) {
28         // (#2.3) Lakukan get request parameter username
29         String username = request.getParameter(name: "username");
30         // (#2.4) Lakukan get request parameter password
31         String password = request.getParameter(name: "password");
32         // (#2.5) Lakukan pengecekan pada if, jika var username bernilai "admin", dan var password bernilai "1234"
33         if ("admin".equals(anObject: username) && "1234".equals(anObject: password)) {
34             HttpSession session = request.getSession();
35             // (#2.6) Lakukan set atribut session ke "user" dari var username sebelumnya
36             session.setAttribute(name: "user", value: username);
37             response.sendRedirect(request.getContextPath() + "/product?menu=view");
38         } else {
39             response.sendRedirect(request.getContextPath() + "/index.jsp?error=1");
40         }
41     } else if ("logout".equals(anObject: action)) {
42         HttpSession session = request.getSession(create: false);
43         // (#2.7) Lakukan pengecekan jika session tidak bernilai null
44         if (session!=null) {
45             // (#2.8) Lakukan pengecekan jika session invalidate dengan memanggil fungsi bawaannya
46             session.invalidate();
47         }
48         response.sendRedirect(request.getContextPath() + "/index.jsp");
49     } else {
50         response.sendRedirect(request.getContextPath() + "/index.jsp");
51     }
52 }

```

3.1

3.2

3.3

3.4

3.5

```

18 protected void doGet(HttpServletRequest request, HttpServletResponse response)
19     throws ServletException, IOException {
20     // (#3.1) Inisialisasi object HttpSession untuk melakukan request session
21     HttpSession session = request.getSession();
22     // (#3.2) Lakukan pengecekan apakah var session untuk atribut "user" bernilai null
23     if (session.getAttribute(name: "user") == null) {
24         // (#3.3) Lakukan send redirect secara langsung ke index.jsp
25         response.sendRedirect(location: "index.jsp");
26         return;
27     }
28
29     String menu = request.getParameter(name: "menu");
30     // (#3.4) Lakukan pengecekan jika var menu bernilai null atau kosong
31     if (menu == null || "".equals(anObject: menu)) {
32         // (#3.5) Lakukan send redirect ke view, gunakan product?
33         response.sendRedirect(location: "view?menu=product");
34         return;
35     }
36 }

```

3.6

3.7

3.8

3.9

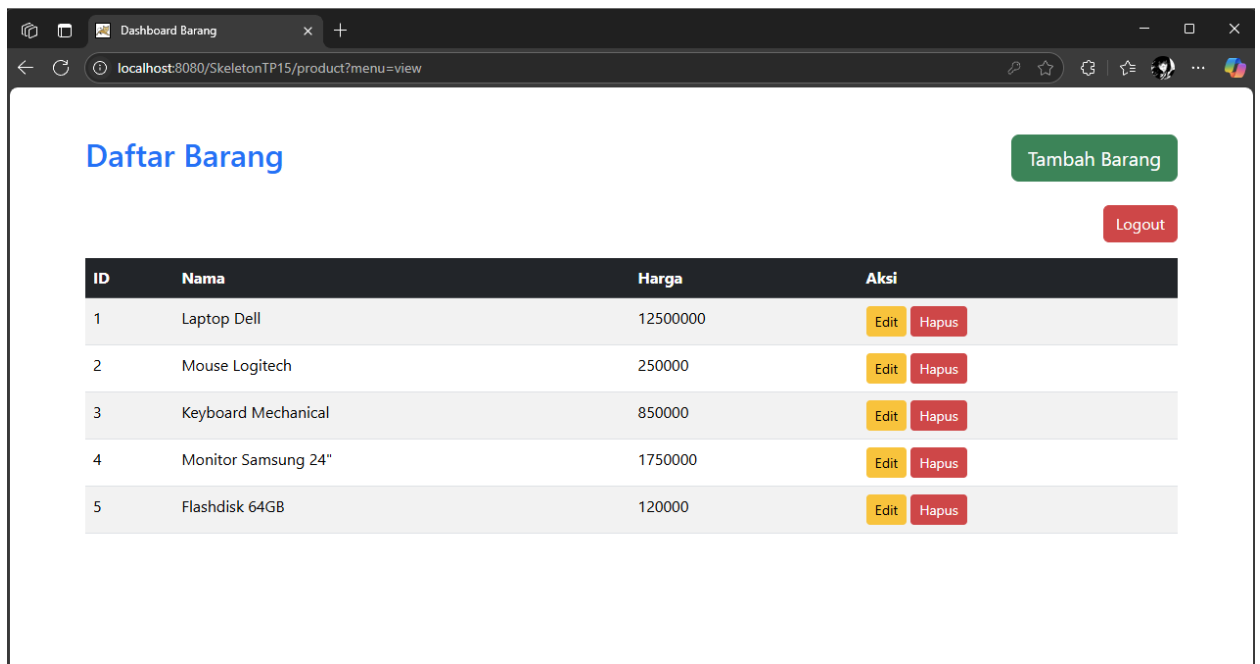
```

91 // (#3.6) Lakukan pengecekan apakah var action bernilai "delete"
92 } else if ("delete".equals(anObject: action)) {
93 // (#3.7) var id menyimpan nilai request parameter "id", lakukan juga parseInt
94 int id = Integer.parseInt(s: request.getParameter(name: "id"));
95 // (#3.8) Dari object Product yang telah dibuat sebelumnya, lakukan setId dengan nilai var id
96 productModel.setId(id);
97 // (#3.9) Dari object Product yang telah dibuat sebelumnya, panggil method delete
98 productModel.delete();
99 }
100
101 response.sendRedirect(location: "product?menu=view");
102 }
103 }

```

3. Pastikan sudah bisa login ke dalam aplikasi!

The screenshot shows a web browser window with the address bar displaying `localhost:8080/SkeletonTP15/index.jsp`. The page contains a login form with a blue header bar labeled "Login". Below the header, there are two input fields: "Username" with the value "admin" and "Password" with the value "1234". A blue "Login" button is positioned below the password field. The browser's address bar and tabs are visible at the top.



4. Hapus salah satu barang pada aplikasi, dan lakukan screenshot bahwa data sudah terhapus!

ID	Nama	Harga	Aksi
1	Laptop Dell	12500000	Edit Hapus
2	Mouse Logitech	250000	Edit Hapus
3	Keyboard Mechanical	850000	Edit Hapus
4	Monitor Samsung 24"	1750000	Edit Hapus
5	Flashdisk 64GB	120000	Edit Hapus

ID	Nama	Harga	Aksi
2	Mouse Logitech	250000	Edit Hapus
3	Keyboard Mechanical	850000	Edit Hapus
4	Monitor Samsung 24"	1750000	Edit Hapus
5	Flashdisk 64GB	120000	Edit Hapus