

**Nama** : Alipiani Dwi Putri  
**NIM** : 312410691  
**Kelas** : TI.24A2  
**Mata Kuliah** : Pemrograman Web 1  
**Dosen pengampu** : Agung Nugroho, S.Kom., M.Kom.

---

<b>UAS Pemrograman Web</b>
----------------------------

<b>SISTEM MANAJEMEN BARANG</b>
--------------------------------

## LINKS & RESOURCES

- GitHub Repository:  
[https://github.com/Alipianidwiputri/project\\_uas\\_web](https://github.com/Alipianidwiputri/project_uas_web)
  - Video Demo YouTube:  
<https://youtu.be/IhNrN6Q6hcA?feature=shared>
  - README Documentation:  
[https://github.com/Alipianidwiputri/project\\_uas\\_web/blob/main/README.md](https://github.com/Alipianidwiputri/project_uas_web/blob/main/README.md)
  - Local URL:  
[http://localhost/project\\_uas\\_web](http://localhost/project_uas_web)
- 

## 1. PROJECT OVERVIEW

Sistem Manajemen Barang adalah aplikasi web untuk mengelola data inventori dengan fitur CRUD lengkap, login, search, filter, dan pagination.

### TEKNOLOGI:

- PHP Native (tanpa framework) dengan MVC pattern
- MySQL database
- Bootstrap 5 untuk UI
- JavaScript untuk interactivity

### URL LOKAL:

[http://localhost/project\\_uas\\_web](http://localhost/project_uas_web)

---

## 2. TECH STACK

### BACKEND:

- PHP 7.4+ : Backend logic, business layer
- MySQL 5.7+ : Database storage
- PDO : Database connection dengan prepared statements
- Apache 2.4+ : Web server

## **FRONTEND:**

- Bootstrap 5.3.0 : Framework CSS, responsive grid
- Bootstrap Icons 1.10.0 : Icon library
- JavaScript ES6 : Form validation, interactivity

## **DATABASE:**

- Name: latihan1
  - Tables: users (3 rows), data\_barang (36 rows)
- 

### **3. FITUR APLIKASI**

#### **a. LOGIN & LOGOUT**

Session-based authentication dengan 2 role: Admin (full access) dan User (read-only). Admin bisa CRUD, User hanya bisa view.

#### **b. DASHBOARD**

Menampilkan statistik total barang dan total user, tabel distribusi barang per kategori, informasi user yang login, dan role badge.

#### **c. CRUD BARANG (Admin Only)**

- Create: Form tambah barang dengan upload gambar (max 2MB)
- Read: Lihat semua barang dengan gambar produk
- Update: Edit data barang existing
- Delete: Hapus barang dengan konfirmasi popup

#### **d. SEARCH BARANG**

Pencarian barang berdasarkan nama menggunakan LIKE query untuk partial match. Real-time search dengan tombol Filter.

#### **e. FILTER KATEGORI**

Filter barang berdasarkan kategori dengan dropdown yang diisi dinamis dari database. Bisa dikombinasikan dengan search. Ada tombol Reset untuk clear semua filter.

#### **f. PAGINATION**

Menampilkan 10 items per halaman dengan navigasi Previous, Next, dan page numbers. Filter dan search tetap aktif saat pindah halaman. Info total halaman dan total data ditampilkan.

#### **g. UPLOAD GAMBAR**

Support format JPG, PNG, JPEG, GIF dengan ukuran maksimal 2MB. Ada preview gambar sebelum upload dan thumbnail di list barang.

## h. RESPONSIVE DESIGN

Mobile: Hamburger menu, sidebar slide, table scroll horizontal

Tablet & Desktop: Full sidebar dengan menu lengkap

Breakpoints: 768px, 992px, 1200px

## i. AUTHORIZATION

Admin memiliki full access ke semua fitur CRUD. User hanya bisa view data, tidak ada tombol Edit/Delete di interface, dan URL admin page di-protect dengan checkAdmin().

---

## 4. STRUKTUR CODE

```
project_uas_web/
├── .htaccess           # URL rewriting
├── index.php           # Entry point & routing
├── config/
│   └── database.php    # Database connection (Singleton)
├── app/
│   ├── controllers/
│   │   ├── AuthController.php  # Login, logout, session
│   │   ├── BarangController.php # CRUD barang
│   │   └── DashboardController.php # Dashboard statistics
│   ├── models/
│   │   ├── User.php           # User queries
│   │   └── Barang.php         # Barang queries
│   └── views/
│       ├── layouts/
│       │   ├── header.php      # Sidebar navigation
│       │   └── footer.php      # Scripts & closing tags
│       ├── auth/
│       │   └── login.php        # Login form
│       ├── dashboard/
│       │   └── index.php        # Dashboard page
│       └── barang/
│           ├── index.php        # List barang
│           ├── create.php       # Form tambah
│           └── edit.php         # Form edit
└── public/
    ├── css/style.css          # Custom styles
    ├── js/script.js           # Custom JavaScript
    └── images/barang/         # Upload folder (36 images)
```

- PATTERN: MVC (Model-View-Controller)
- Model: Database logic

- View: HTML presentation
  - Controller: Business logic
- 

## 5. PENJELASAN CODE UTAMA

### 5.1 ROUTING (index.php)

**FUNGSI:** Entry point aplikasi, handle routing ke controller

```
<?php
session_start();

define('BASE_PATH', __DIR__);
define('BASE_URL', 'http://localhost/project_uas_web');

// Autoload classes
spl_autoload_register(function($class) {
    $paths = [
        BASE_PATH . '/app/controllers/' . $class . '.php',
        BASE_PATH . '/app/models/' . $class . '.php',
        BASE_PATH . '/config/' . $class . '.php'
    ];

    foreach ($paths as $path) {
        if (file_exists($path)) {
            require_once $path;
            break;
        }
    }
});

// Parse URL
$url = isset($_GET['url']) ? $_GET['url'] : 'auth/login';
$url = explode('/', $url);

// Route to controller
$controllerName = ucfirst($url[0]) . 'Controller';
$controller = new $controllerName();

$methodName = isset($url[1]) ? $url[1] : 'index';
$params = isset($url[2]) ? array_slice($url, 2) : [];

// Execute
call_user_func_array([$controller, $methodName], $params);
?>
```

## CARA KERJA:

URL: /barang/edit/5

- Controller: BarangController
- Method: edit()
- Params: [5]
- Execute: BarangController->edit(5)

## 5.2 DATABASE CONNECTION (config/database.php)

**FUNGSI:** Koneksi database dengan Singleton pattern

```
<?php
define('DB_HOST', 'localhost');
define('DB_USER', 'root');
define('DB_PASS', '');
define('DB_NAME', 'latihan1');

class Database {
    private static $instance = null;
    private $conn;

    private function __construct() {
        try {
            $this->conn = new PDO(
                "mysql:host=" . DB_HOST . ";dbname=" . DB_NAME,
                DB_USER,
                DB_PASS
            );
            $this->conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        } catch(PDOException $e) {
            die("Connection failed: " . $e->getMessage());
        }
    }

    public static function getInstance() {
        if (self::$instance == null) {
            self::$instance = new Database();
        }
        return self::$instance;
    }

    public function getConnection() {
        return $this->conn;
    }
}
?>
```

**KEUNTUNGAN:** Hanya 1 koneksi database untuk seluruh aplikasi, efisien.

### 5.3 AUTHENTICATION (app/controllers/AuthController.php)

**FUNGSI:** Handle login, logout, session, authorization

```
<?php
class AuthController {
    private $userModel;

    public function __construct() {
        $this->userModel = new User();
    }

    public function login() {
        if (isset($_SESSION['user_id'])) {
            header('Location: ' . BASE_URL . '/dashboard');
            exit;
        }

        if ($_SERVER['REQUEST_METHOD'] === 'POST') {
            $username = trim($_POST['username']);
            $password = trim($_POST['password']);

            $user = $this->userModel->getUserByUsername($username);

            if ($user && $this->userModel->verifyPassword($password, $user['password'])) {
                $_SESSION['user_id'] = $user['id'];
                $_SESSION['username'] = $user['username'];
                $_SESSION['nama'] = $user['nama'];
                $_SESSION['role'] = ($user['username'] === 'admin') ? 'admin' : 'user';

                header('Location: ' . BASE_URL . '/dashboard');
                exit;
            } else {
                $error = "Username atau password salah!";
            }
        }

        require_once BASE_PATH . '/app/views/auth/login.php';
    }

    public function logout() {
        session_unset();
        session_destroy();
        header('Location: ' . BASE_URL . '/auth/login');
        exit;
    }
}
```

```

public static function checkLogin() {
    if (!isset($_SESSION['user_id'])) {
        header('Location: ' . BASE_URL . '/auth/login');
        exit;
    }
}

public static function checkAdmin() {
    self::checkLogin();
    if ($_SESSION['role'] !== 'admin') {
        $_SESSION['error'] = "Akses ditolak! Anda bukan admin.";
        header('Location: ' . BASE_URL . '/dashboard');
        exit;
    }
}
}
?>

```

## ROLE LOGIC:

- username === 'admin' → role = 'admin' (full access)
- username !== 'admin' → role = 'user' (read-only)

## 5.4 CRUD BARANG (app/controllers/BarangController.php)

**FUNGSI:** Handle operasi CRUD barang

```

<?php
class BarangController {
    private $barangModel;

    public function __construct() {
        AuthController::checkLogin();
        $this->barangModel = new Barang();
    }

    // READ: List barang with pagination, search, filter
    public function index() {
        $page = isset($_GET['page']) ? (int)$_GET['page'] : 1;
        $search = isset($_GET['search']) ? trim($_GET['search']) : "";
        $kategori = isset($_GET['kategori']) ? trim($_GET['kategori']) : "";

        $limit = 10;
        $offset = ($page - 1) * $limit;

        $barang = $this->barangModel->getAllBarang($page, $limit, $search, $kategori);
        $totalBarang = $this->barangModel->countBarang($search, $kategori);
    }
}

```

```

$totalPages = ceil($totalBarang / $limit);
$kategoriList = $this->barangModel->getKategoriList();

require_once BASE_PATH . '/app/views/barang/index.php';
}

// CREATE: Form tambah barang
public function create() {
    AuthController::checkAdmin();

    if ($_SERVER['REQUEST_METHOD'] === 'POST') {
        // Validate input
        $errors = [];
        if (empty(trim($_POST['nama']))) {
            $errors[] = "Nama barang wajib diisi!";
        }

        // Handle image upload
        if (isset($_FILES['gambar']) && $_FILES['gambar']['error'] === UPLOAD_ERR_OK) {
            $uploadDir = BASE_PATH . '/public/images/barang/';
            $fileName = $_FILES['gambar']['name'];
            $fileExt = strtolower(pathinfo($fileName, PATHINFO_EXTENSION));
            $allowedExts = ['jpg', 'jpeg', 'png', 'gif'];

            if (in_array($fileExt, $allowedExts) && $_FILES['gambar']['size'] <= 2097152) {
                $newFileName = 'barang_' . time() . '.' . $fileExt;
                move_uploaded_file($_FILES['gambar']['tmp_name'], $uploadDir . $newFileName);
                $gambar = $newFileName;
            }
        }

        // Save to database
        if (empty($errors)) {
            $data = [
                'nama' => trim($_POST['nama']),
                'kategori' => trim($_POST['kategori']),
                'harga_beli' => (float)$_POST['harga_beli'],
                'harga_jual' => (float)$_POST['harga_jual'],
                'stok' => (int)$_POST['stok'],
                'gambar' => $gambar ?? null
            ];

            if ($this->barangModel->createBarang($data)) {
                $_SESSION['success'] = "Barang berhasil ditambahkan!";
                header('Location: ' . BASE_URL . '/barang');
                exit;
            }
        }
    }
}

```



```

    }

    require_once BASE_PATH . '/app/views/barang/create.php';
}

// UPDATE: Edit barang
public function edit($id) {
    AuthController::checkAdmin();

    $barang = $this->barangModel->getBarangById($id);

    if ($_SERVER['REQUEST_METHOD'] === 'POST') {
        // Update logic similar to create
        if ($this->barangModel->updateBarang($id, $data)) {
            $_SESSION['success'] = "Barang berhasil diupdate!";
            header('Location: ' . BASE_URL . '/barang');
            exit;
        }
    }
}

require_once BASE_PATH . '/app/views/barang/edit.php';
}

// DELETE: Hapus barang
public function delete($id) {
    AuthController::checkAdmin();

    $barang = $this->barangModel->getBarangById($id);

    if ($barang) {
        // Delete image file
        if (!empty($barang['gambar'])) {
            $imagePath = BASE_PATH . '/public/images/barang/' . $barang['gambar'];
            if (file_exists($imagePath)) {
                unlink($imagePath);
            }
        }

        // Delete from database
        if ($this->barangModel->deleteBarang($id)) {
            $_SESSION['success'] = "Barang berhasil dihapus!";
        }
    }

    header('Location: ' . BASE_URL . '/barang');
    exit;
}
}

```

?>

## KEY POINTS:

- Authorization check di setiap method (checkAdmin untuk CRUD)
- Upload gambar dengan validasi format dan size
- Feedback message via session flash
- Redirect after POST (PRG pattern)

## 5.5 MODEL BARANG (app/models/Barang.php)

**FUNGSI:** Query database untuk CRUD barang

```
<?php
class Barang {
    private $db;

    public function __construct() {
        $this->db = Database::getInstance()->getConnection();
    }

    // Get all barang with search, filter, pagination
    public function getAllBarang($page, $limit, $search = "", $kategori = "") {
        $offset = ($page - 1) * $limit;

        $sql = "SELECT * FROM data_barang WHERE 1=1";
        $params = [];

        if (!empty($search)) {
            $sql .= " AND nama LIKE :search";
            $params[':search'] = '%' . $search . '%';
        }

        if (!empty($kategori)) {
            $sql .= " AND kategori = :kategori";
            $params[':kategori'] = $kategori;
        }

        $sql .= " ORDER BY id DESC LIMIT :offset, :limit";

        $stmt = $this->db->prepare($sql);

        foreach ($params as $key => $value) {
            $stmt->bindValue($key, $value);
        }

        $stmt->bindValue(':offset', $offset, PDO::PARAM_INT);
        $stmt->bindValue(':limit', $limit, PDO::PARAM_INT);
    }
}
```

```

$stmt->execute();
return $stmt->fetchAll();
}

// Count total barang
public function countBarang($search = "", $kategori = "") {
    $sql = "SELECT COUNT(*) as total FROM data_barang WHERE 1=1";
    $params = [];

    if (!empty($search)) {
        $sql .= " AND nama LIKE :search";
        $params[':search'] = '%' . $search . '%';
    }

    if (!empty($kategori)) {
        $sql .= " AND kategori = :kategori";
        $params[':kategori'] = $kategori;
    }

    $stmt = $this->db->prepare($sql);
    $stmt->execute($params);
    return $stmt->fetch()['total'];
}

// Get barang by ID
public function getBarangById($id) {
    $stmt = $this->db->prepare("SELECT * FROM data_barang WHERE id = ?");
    $stmt->execute([$id]);
    return $stmt->fetch();
}

// Create barang
public function createBarang($data) {
    $sql = "INSERT INTO data_barang
        (nama, kategori, harga_beli, harga_jual, stok, gambar)
        VALUES (:nama, :kategori, :harga_beli, :harga_jual, :stok, :gambar)";

    $stmt = $this->db->prepare($sql);
    return $stmt->execute($data);
}

// Update barang
public function updateBarang($id, $data) {
    $sql = "UPDATE data_barang SET
        nama = :nama, kategori = :kategori,
        harga_beli = :harga_beli, harga_jual = :harga_jual,
        stok = :stok, gambar = :gambar
    ";
}

```

```

        WHERE id = :id";

        $data['id'] = $id;
        $stmt = $this->db->prepare($sql);
        return $stmt->execute($data);
    }

    // Delete barang
    public function deleteBarang($id) {
        $stmt = $this->db->prepare("DELETE FROM data_barang WHERE id = ?");
        return $stmt->execute([$id]);
    }

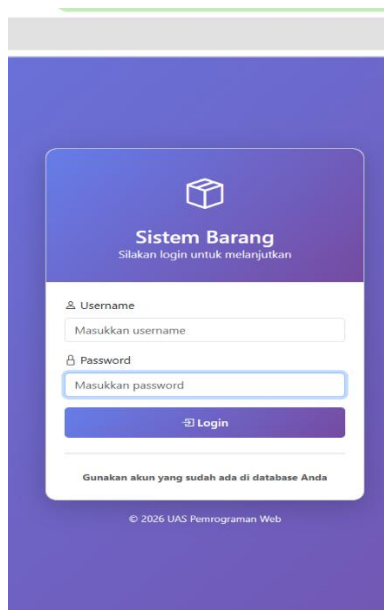
    // Get kategori list for filter
    public function getKategoriList() {
        $stmt = $this->db->query("SELECT DISTINCT kategori FROM data_barang ORDER BY
kategori");
        return $stmt->fetchAll();
    }
}
?>

```

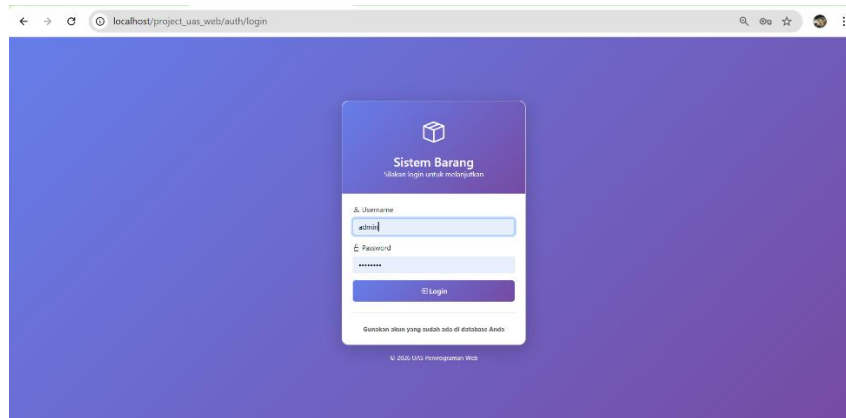
**SECURITY:** Semua query pakai prepared statements untuk mencegah SQL injection.

## 6. SCREENSHOT APLIKASI

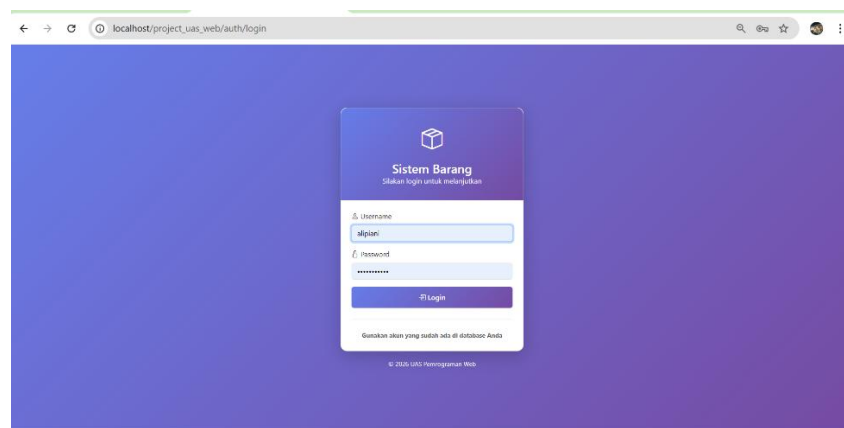
### 6.1 AUTHENTICATION



Halaman login dengan gradient design modern, form input username & password, dan tombol login.

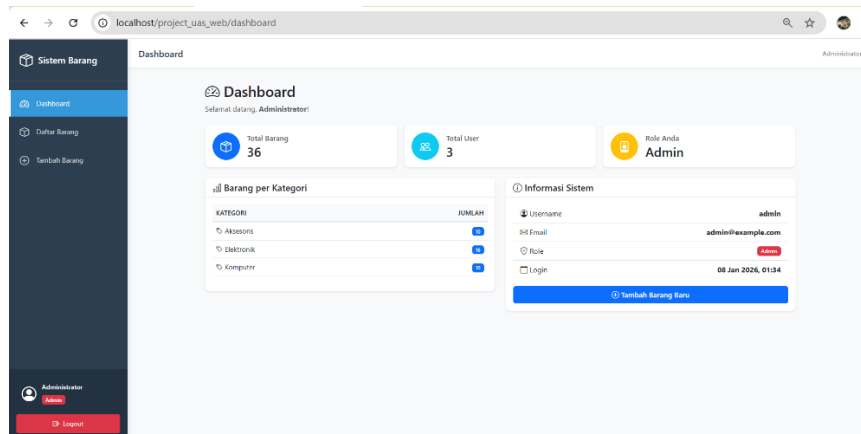


Form login terisi dengan username: admin, password: ..... (hidden).

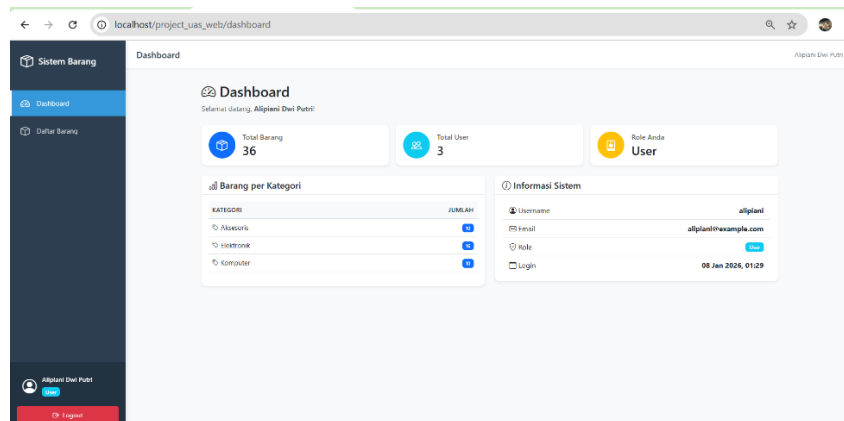


Form login terisi dengan username: alipiani, password: ..... (hidden).

## 6.2 DASHBOARD

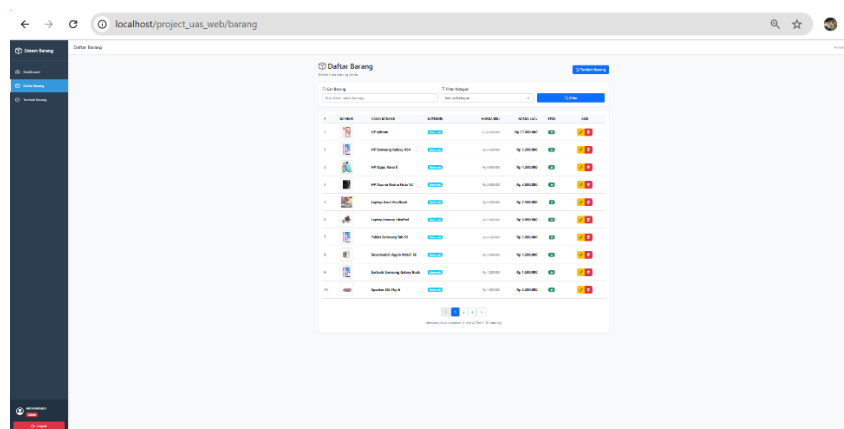


Sidebar kiri dengan 3 menu (Dashboard, Daftar Barang, Tambah Barang), statistik card menampilkan Total Barang (36), Total User (3), Role (Admin), tabel distribusi barang per kategori, dan user info di sidebar.

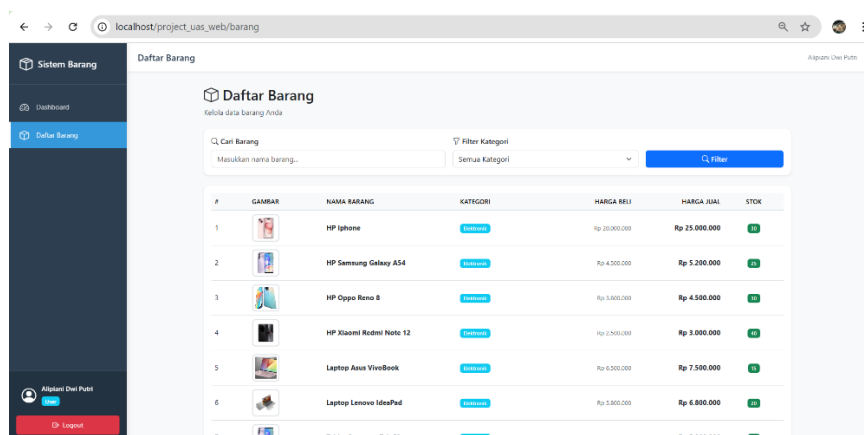


Sidebar hanya 2 menu (Dashboard, Daftar Barang), tidak ada menu Tambah Barang, role badge berwarna biru (User).

## 6.3 DAFTAR BARANG

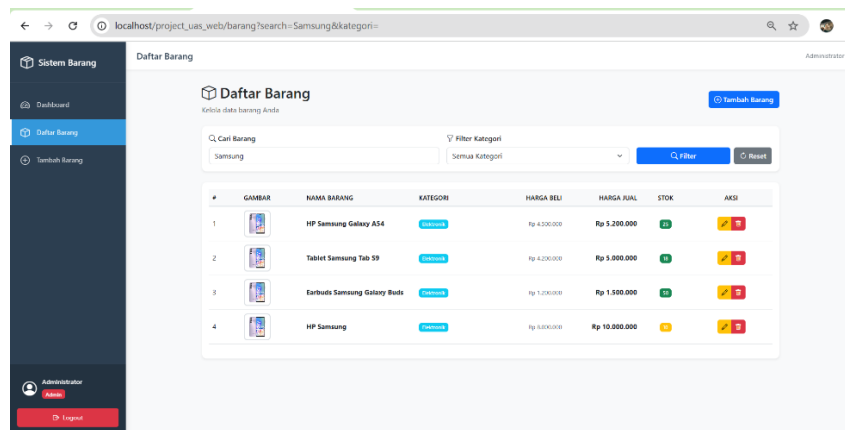


Tabel dengan kolom: #, Gambar, Nama, Kategori, Harga Beli, Harga Jual, Stok, Aksi (Edit & Hapus). Search box dan filter dropdown di atas. Pagination di bawah.

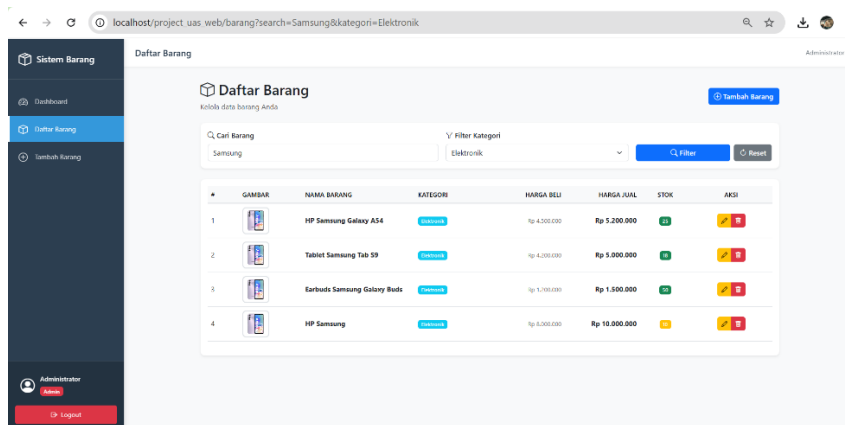


Tabel sama seperti admin tetapi tidak ada kolom Aksi dan tidak ada tombol Edit/Hapus. User hanya bisa view data.

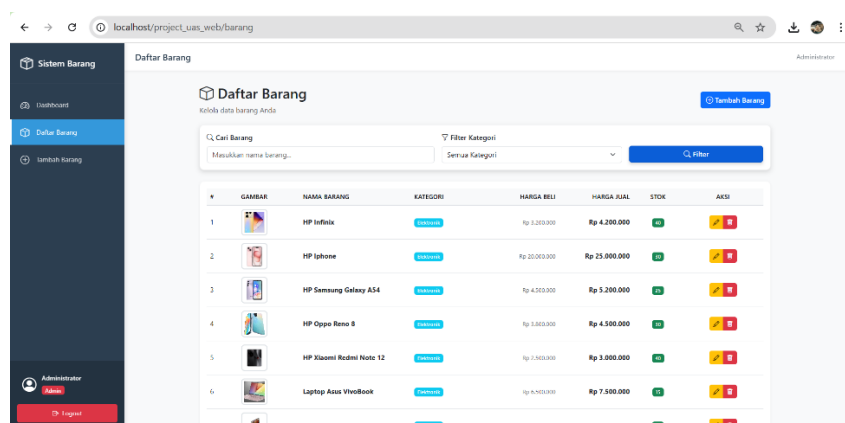
## 6.4 SEARCH & FILTER



Input search diisi "samsung", hasil menampilkan barang dengan nama mengandung "samsung", tombol Reset muncul.



Search "hp" + Filter "Elektronik", hasil menampilkan barang yang nama mengandung "hp" DAN kategori Elektronik.



Setelah klik tombol Reset, tampilan kembali ke semua data tanpa filter.

## 6.5 CRUD OPERATIONS (ADMIN ONLY)

The screenshot shows the 'Tambah Barang Baru' form in the Admin interface. The form is titled 'Tambah Barang Baru' and includes a subtitle 'Lengkapi form di bawah untuk menambahkan barang'. The form fields are:

- Nama Barang \***: Input field with example 'Contoh: HP Samsung'.
- Kategori \***: Input field with example 'Contoh: Elektronik'.
- Harga Beli \***: Input field with example 'Rp. 0'.
- Harga Jual \***: Input field with example 'Rp. 0'.
- Stok \***: Input field with example '0'.
- Upload Gambar Produk**: Section with a 'Choose file' button and a note 'Format: JPG, PNG, JPEG (Max 2MB). Atau ketik nama file manual di bawah'.
- Atau ketik nama file gambar:**: Input field with example 'Contoh: hp\_samsung.jpg'.

At the bottom of the form are two buttons: 'Simpan Barang' (Save Item) and 'Batal' (Cancel).

Form dengan field: Nama Barang, Kategori, Harga Beli, Harga Jual, Stok, Upload Gambar. Tombol Simpan Barang dan Batal.

The screenshot shows the 'Tambah Barang Baru' form with example data filled in. The form fields are:

- Nama Barang \***: Input field with example 'Keyboard Gaming RGB'.
- Kategori \***: Input field with example 'Aksesoris'.
- Harga Beli \***: Input field with example 'Rp. 500000'.
- Harga Jual \***: Input field with example 'Rp. 750000'.
- Stok \***: Input field with example '10'.
- Upload Gambar Produk**: Section with a 'Choose file' button and a note 'Format: JPG, PNG, JPEG (Max 2MB). Atau ketik nama file manual di bawah'.
- Atau ketik nama file gambar:**: Input field with example 'Keyboard.jpg'.

At the bottom of the form are two buttons: 'Simpan Barang' (Save Item) and 'Batal' (Cancel). A preview of the uploaded image is shown below the file upload section.

Semua field terisi dengan contoh: Keyboard Gaming RGB, Aksesoris, 500000, 750000, 10. Preview gambar muncul setelah upload.

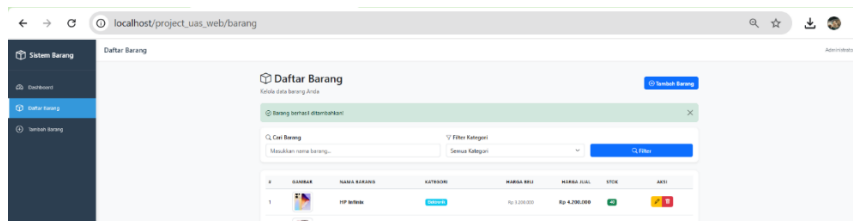
The screenshot shows the 'Edit Barang' form in the Admin interface. The form is titled 'Edit Barang' and includes a subtitle 'Update informasi barang'. The form fields are:

- Nama Barang \***: Input field with example 'HP Iphone'.
- Kategori \***: Input field with example 'Elektronik'.
- Harga Beli \***: Input field with example 'Rp. 2000000.00'.
- Harga Jual \***: Input field with example 'Rp. 2500000.00'.
- Stok \***: Input field with example '30'.
- Upload Gambar Produk**: Section with a 'Choose file' button and a note 'Format: JPG, PNG, JPEG (Max 2MB) atau unggah foto dari galeri perangkat'.
- Atau ketik nama file gambar:**: Input field with example 'HP Iphone.jpg'.

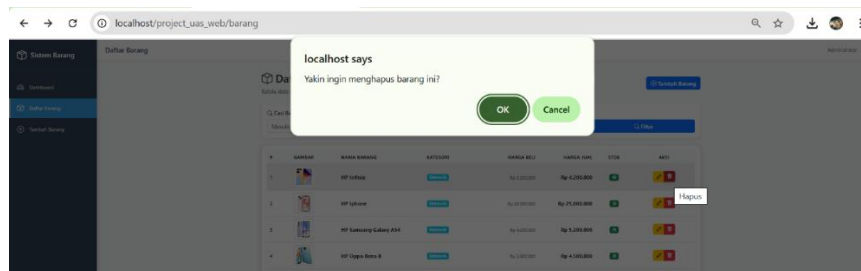
At the bottom of the form are two buttons: 'Update Barang' (Update Item) and 'Batal' (Cancel). A preview of the uploaded image is shown below the file upload section.

Form terisi dengan data existing, info tanggal input ditampilkan, preview gambar saat ini, tombol Update Barang dan Batal.



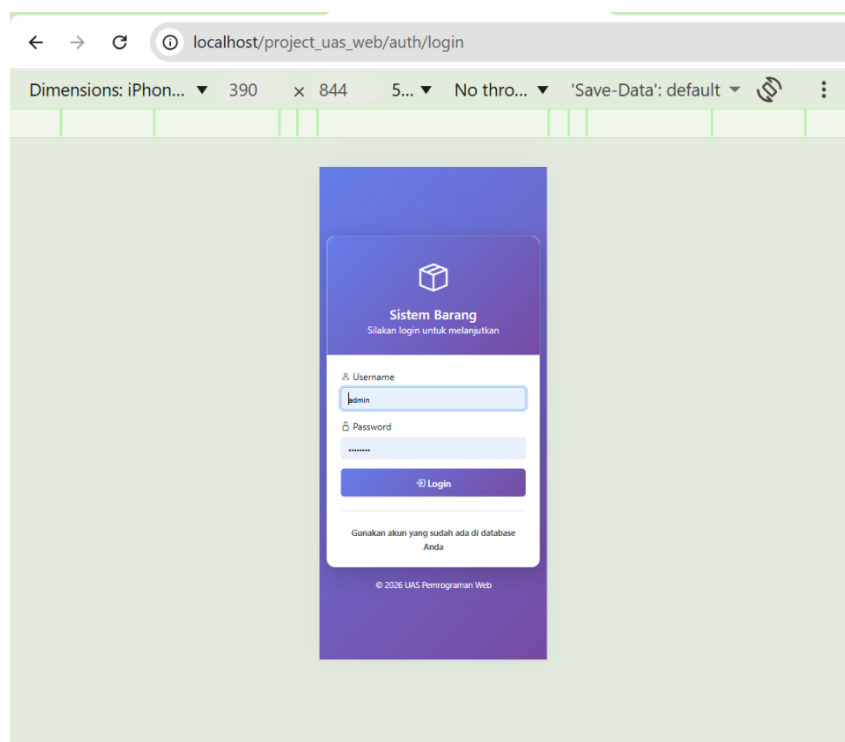


Alert hijau menampilkan "Barang berhasil ditambahkan!" atau "Barang berhasil diupdate!". Auto-hide setelah 5 detik.

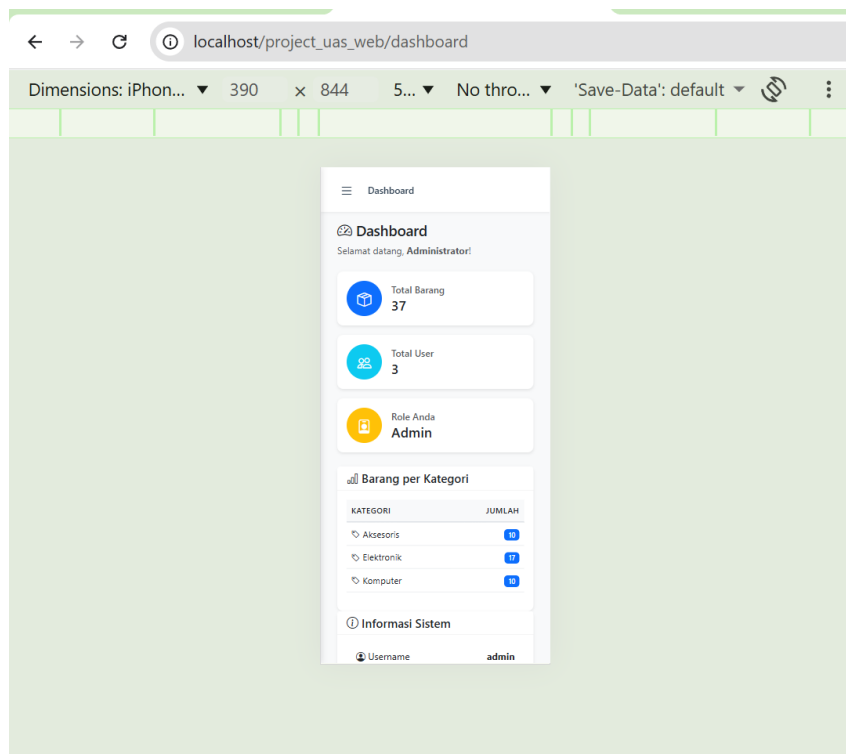


Modal popup konfirmasi dengan text "Yakin ingin menghapus barang ini?", tombol Ya Hapus (merah) dan Batal (abu-abu).

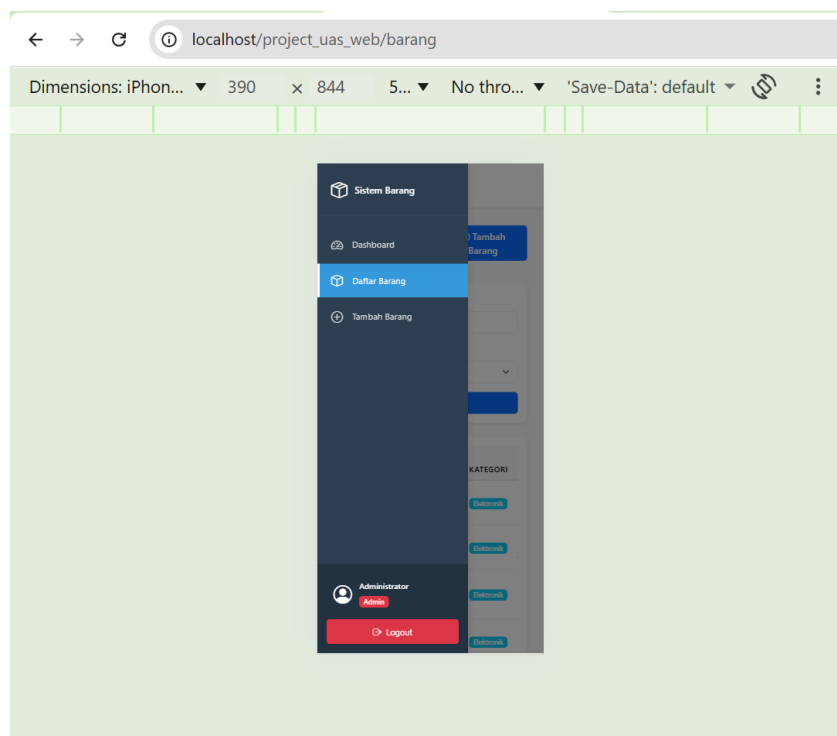
## 6.6 RESPONSIVE MOBILE VIEW



Login page dalam mode mobile (iPhone 12 Pro), layout vertikal, form full-width.



Hamburger menu ( $\equiv$ ) di navbar atas, card statistik stack vertikal, table responsive dengan horizontal scroll.



Hamburger menu diklik, sidebar slide dari kiri dengan overlay backdrop, menu list ditampilkan.

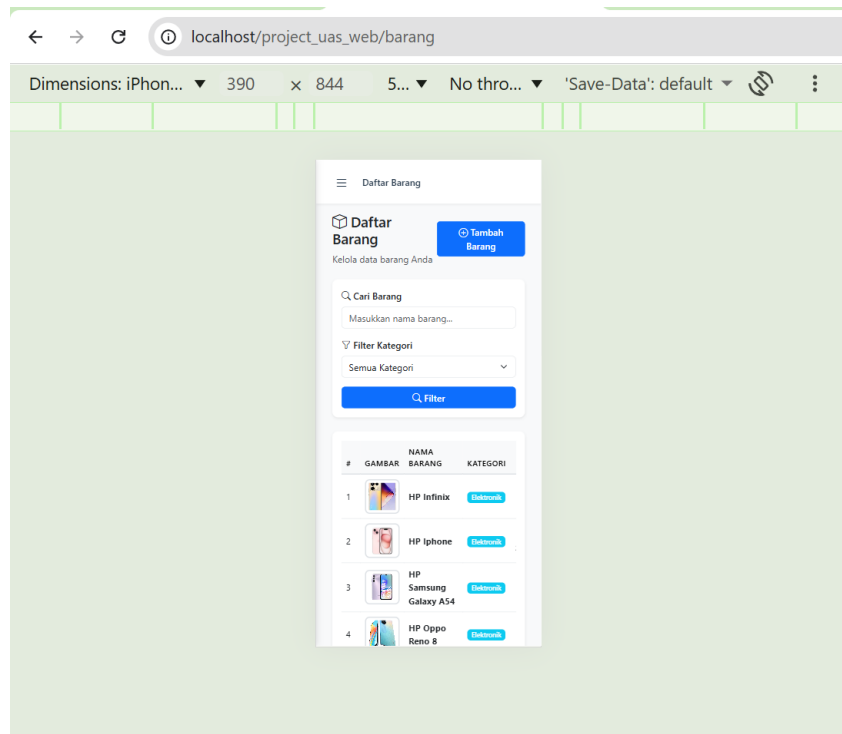


Table responsive dengan horizontal scroll, button touch-friendly, filter dan search stack vertikal.

---

## 7. INSTALASI & SETUP

### LANGKAH INSTALASI:

#### 1. CLONE REPOSITORY

git clone <https://github.com/Alipianidwiputri/LabWeb13.git>  
atau download ZIP → extract ke C:\xampp\htdocs\project\_uas\_web

#### 2. DATABASE SETUP

Database sudah ada: latihan1  
Tabel sudah ada: users, data\_barang  
Data sample sudah tersedia (3 users, 36 barang)

#### 3. KONFIGURASI (Opsional)

Edit config/database.php jika setting berbeda:  
DB\_HOST: localhost  
DB\_USER: root  
DB\_PASS: (kosong)  
DB\_NAME: latihan1

#### 4. ENABLE MOD\_REWRITE

Buka C:\xampp\apache\conf\httpd.conf  
Uncomment: LoadModule rewrite\_module modules/mod\_rewrite.so

Ganti AllowOverride None → AllowOverride All  
Restart Apache

## 5. AKSES APLIKASI

[http://localhost/project\\_uas\\_web](http://localhost/project_uas_web)

### TROUBLESHOOTING:

1. 404 Not Found → Cek mod\_rewrite sudah enable, .htaccess ada di root
2. Database Connection Failed → Cek MySQL running, cek credentials
3. CSS Tidak Load → Hard refresh (Ctrl+F5), cek path di header.php
4. Upload Gagal → Cek folder public/images/barang/ ada, permission 755

---

## 8. LOGIN CREDENTIALS

### ADMIN (Full Access):

**Username:** admin

**Password:** admin123

**Hak Akses:** View Dashboard, View Daftar Barang, Create/Edit/Delete Barang, Search & Filter, Upload Gambar

### USER (Read Only):

**Username:** alipiani

**Password:** password123

**Hak Akses:** View Dashboard, View Daftar Barang, Search & Filter saja

**Tidak bisa:** Create, Edit, Delete barang

### USER 2 (Read Only):

**Username:** user

**Password:** user123

**Hak Akses:** Sama seperti alipiani