

PERTEMUAN IV

PROGRESS INVENTORY MANAGEMENT

- Memberikan arahan dan panduan ke beberapa AI, yaitu chatgpt, qwen, grox, dll.

Pada awalnya saya mencoba membuat inventory management menggunakan chatgpt dan Qwen tetapi percobaan saya masi terdapat kendala seperti mengalami eror dalam kode sql maupun ketika sudah tidak mengalami eror tetapi localhost yang kita buat tidak muncul. Berikut saya lampirkan kode percobaan saya.

Instruksi yang saya cari sesuai dengan permintaan saya

1. Chatgpt

```
MySQL Database (salon_inventory.sql)

sql

CREATE DATABASE salon_inventory;
USE salon_inventory;

CREATE TABLE produk (
  id INT AUTO_INCREMENT PRIMARY KEY,
  nama_produk VARCHAR(100),
  kategori VARCHAR(50),
  harga_beli DECIMAL(10,2),
  harga_jual DECIMAL(10,2),
  stok INT,
  satuan VARCHAR(20)
);
```

```
style.css (tema pink + animasi)

css

body {
  font-family: 'Segoe UI', sans-serif;
  background: #ffe0f0;
  color: #333;
  margin: 0;
  padding: 0;
}

header {
  background: #ff69b4;
  padding: 20px;
  text-align: center;
  color: white;
  font-size: 24px;
  animation: slideIn 1s ease-out;
}


@keyframes slideIn {
  from {transform: translateY(-100%);}
  to {transform: translateY(0);}
}

.container {
  width: 90%;
  margin: 20px auto;
}

.card {
  background: white;
  padding: 20px;
  border-radius: 12px;
  box-shadow: 0 4px 12px rgba(0,0,0,0.1);
  margin-bottom: 20px;
  transition: 0.3s ease;
}

.card:hover {
  transform: scale(1.02);
}
```


Save Edit



php/db.php

```
php
<?php
$conn = new mysqli("localhost", "root", "", "salon_inventory");
if ($conn->connect_error) {
    die("Koneksi gagal: " . $conn->connect_error);
}
?>
```

+ php/add_item.php

```
php
<?php
include 'db.php';
$data = json_decode(file_get_contents("php://input"), true);
$sql = "INSERT INTO produk (nama_produk, kategori, harga_beli, harga_jual, stok, satuan)
VALUES (?, ?, ?, ?, ?, ?)";
$stmt = $conn->prepare($sql);
$stmt->bind_param("ssdis", $data['nama_produk'], $data['kategori'], $data['harga_beli'],
    $data['harga_jual'], $data['stok'], $data['satuan']);
$stmt->execute();
```

php/get_items.php

```
php
<?php
include 'db.php';
$result = $conn->query("SELECT * FROM produk");
$data = [];
while($row = $result->fetch_assoc()) {
    $data[] = $row;
}
echo json_encode($data);
```

✗ php/delete_item.php

```
php
<?php
include 'db.php';
$id = $_GET['id'];
$conn->query("DELETE FROM produk WHERE id=$id");
```

2. QWEN

Project Structure

The project will be organized into the following folders and files:

```
1 /salon_inventory/
2 |— index.html
3 |— style.css
4 |— script.js
5 |— server.php
6 |— db_config.php
7 |— images/
8 |— icon1.png
9 |— icon2.png
10 |— ...
```

Sample Data

Insert some sample data into the tables:

```
sql                                                                    Copy
1  -- Insert products
2  v INSERT INTO products (name, description, image_url) VALUES
3  ('Hair Shampoo', 'High-quality shampoo for all hair types.', 'images/shampoo.png'),
4  ('Hair Conditioner', 'Moisturizing conditioner for dry hair.', 'images/conditioner.png'),
5  ('Scissors', 'Professional-grade scissors for hair cutting.', 'images/scissors.png'),
6  ('Hair Dryer', 'Powerful hair dryer for styling.', 'images/hair_dryer.png');
7
8  -- Insert stock
9  v INSERT INTO stock (product_id, quantity, purchase_date) VALUES
10 (1, 50, '2023-10-01'),
11 (2, 30, '2023-10-02'),
12 (3, 10, '2023-10-03'),
13 (4, 5, '2023-10-04');
```

Step 2: Frontend Development

index.html

This file will contain the main structure of the application.

```
html                                                                    Copy
1  <!DOCTYPE html>
2  v <html lang="en">
3  v <head>
4  <meta charset="UTF-8">
5  <meta name="viewport" content="width=device-width, initial-scale=1.0">
6  <title>Salon Inventory Management</title>
7  <link rel="stylesheet" href="style.css">
8  </head>
9  v <body>
10 v <div class="container">
11 <!-- Header -->
12 v <header>
13 <h1>KANDI Inventory</h1>
14 v <nav>
15 v <ul>
16 <li><a href="#dashboard">Dashboard</a></li>
```

style.css

This file will define the styles for the application.

```
css
1  /* General Styles */
2  body {
3    font-family: Arial, sans-serif;
4    background-color: #f9f5ff; /* Soft pink background */
5    margin: 0;
6    padding: 0;
7  }
8
9  .container {
10   max-width: 1200px;
11   margin: 0 auto;
12   padding: 20px;
13 }
14
15 header {
16   background-color: #d6bcfa; /* Pinkish header */
17   color: white;
18   padding: 10px 20px;
19   display: flex;
20   justify-content: space-between;
21   align-items: center;
22 }
23
24 nav ul {
25   list-style: none;
26   display: flex;
27   gap: 20px;
28 }
29
30 nav ul li a {
31   text-decoration: none;
32   color: white;
33   font-weight: bold;
34 }
35
36 /* Dashboard Styles */
37 .stats {
38   display: flex;
```

script.js

This file will handle the frontend logic and API calls.

```
javascript
1  // Fetch total products
2  fetch('/api/products/count')
3    .then(response => response.json())
4    .then(data => document.getElementById('total-products').textContent = data.count);
5
6  // Fetch low stock products
7  fetch('/api/stock/low')
8    .then(response => response.json())
9    .then(data => document.getElementById('low-stock').textContent = data.length);
10
11 // Fetch recent purchases
12 fetch('/api/purchases/recent')
13   .then(response => response.json())
14   .then(data => document.getElementById('recent-purchases').textContent = data.length);
15
16 // Load product list
17 document.addEventListener('DOMContentLoaded', () => {
18   fetch('/api/products')
19     .then(response => response.json())
20     .then(products => {
21       const productList = document.getElementById('product-list');
22       products.forEach(product => {
23         const div = document.createElement('div');
24         div.innerHTML = `
25           ${product.name}</strong>
27           <p>${product.description}</p>
28         `;
29         productList.appendChild(div);
30       });
31     });
32
33   // Load stock table
34   fetch('/api/stock')
```

Step 3: Backend Development (PHP)

server.php

This file will handle API endpoints.

```
php
1 <?php
2 require_once 'db_config.php';
3
4 // Get database connection
5 function getDbConnection() {
6     $host = 'localhost';
7     $dbname = 'salon_inventory';
8     $username = 'root';
9     $password = '';
10
11     try {
12         return new PDO("mysql:host=$host;dbname=$dbname", $username, $password);
13     } catch (PDOException $e) {
14         die("Could not connect to the database $dbname :". $e->getMessage());
15     }
16 }
17
18 // API Endpoints
19 if ($_SERVER['REQUEST_METHOD'] === 'GET') {
20     switch ($_GET['route']) {
21         case 'products':
22             echo json_encode(getProducts());
23             break;
24         case 'products/count':
25             echo json_encode(['count' => countProducts()]);
26             break;
27         case 'stock':
28             echo json_encode(getStock());
29             break;
30         case 'stock/low':
31             echo json_encode(getLowStock());
32             break;
33     }
34 }
```

Step 4: Database Configuration (db_config.php)

```
php
1 <?php
2 $pdo = getDbConnection();
3 ?>
```