

Nama : Nisrina Ardyanisa Puspitaningrum
NIM : 12030123120050
Kelas : D
Mata Kuliah : Pengkodean dan Pemrograman

Tugas 3

Aplikasi Persediaan Penjualan

https://tugascodingnisrina.w3spaces-preview.com/persediaan_penjualan.html

Aplikasi Persediaan & Penjualan

Tambah Produk

Nama Produk

Stok

Harga

Tambah

Cari produk...

No	Nama	Stok	Harga	Total	Aksi
----	------	------	-------	-------	------

Total Stok: 0 | Total Penjualan: Rp0

Activate Windows
Go to Settings to activate Windows.

nutella berbagi momen manis kebersamaan BELI SEKARANG

Home HTML HTML Basic Tryit basic HTML document

Run

Result Size: 829 x 615 Get your own website

Aplikasi Persediaan & Penjualan

Tambah Produk

Nama Produk

Stok

Harga

Tambah

Cari produk...

No	Nama	Stok	Harga	Total	Aksi
----	------	------	-------	-------	------

Total Stok: 0 | Total Penjualan: Rp0

Activate Windows
Go to Settings to activate Windows.

<!DOCTYPE html>

```

<html lang="id">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Aplikasi Persediaan & Penjualan</title>

  <!-- W3.CSS -->
  <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css" />

  <!-- Style Internal -->
  <style>
    .logo {
      width: 50px;
      vertical-align: middle;
      margin-right: 10px;
      float: left;
    }
    h2 {
      line-height: 50px;
    }
    #produkTable td, #produkTable th {
      text-align: center;
    }
    button {
      margin-right: 5px;
    }
  </style>
</head>
<body>

  <!-- Logo dan Judul -->
  <div class="w3-container w3-teal w3-padding">
    
    <h2 class="w3-center">Aplikasi Persediaan & Penjualan</h2>
  </div>

  <!-- Form Tambah Produk -->
  <div class="w3-container w3-margin-top">
    <div class="w3-card-4 w3-padding w3-light-grey">
      <h3>Tambah Produk</h3>
      <input class="w3-input w3-border w3-margin-bottom" type="text" id="namaProduk"
placeholder="Nama Produk">
      <input class="w3-input w3-border w3-margin-bottom" type="number" id="stokProduk"
placeholder="Stok">
      <input class="w3-input w3-border w3-margin-bottom" type="number" id="hargaProduk"
placeholder="Harga">

```

```

        <button onclick="tambahProduk()" class="w3-button w3-teal">Tambah</button>
    </div>
</div>

<!-- Pencarian -->
<div class="w3-container w3-margin-top">
    <input class="w3-input w3-border" type="text" id="searchInput" onkeyup="cariProduk()"
placeholder="Cari produk...">
</div>

<!-- Tabel Produk -->
<div class="w3-container w3-margin-top">
    <table class="w3-table-all w3-hoverable" id="produkTable">
        <thead>
            <tr class="w3-teal">
                <th>No</th>
                <th>Nama</th>
                <th>Stok</th>
                <th>Harga</th>
                <th>Total</th>
                <th>Aksi</th>
            </tr>
        </thead>
        <tbody id="tableBody">
            <!-- Data produk akan ditampilkan di sini -->
        </tbody>
    </table>
    <p class="w3-right w3-margin">Total Stok: <span id="totalStok">0</span> | Total Penjualan: <span
id="totalHarga">Rp0</span></p>
</div>

<!-- Script Internal -->
<script>
    let produkList = [];

    function tambahProduk() {
        const nama = document.getElementById("namaProduk").value.trim();
        const stok = parseInt(document.getElementById("stokProduk").value);
        const harga = parseInt(document.getElementById("hargaProduk").value);

        if (!nama || isNaN(stok) || isNaN(harga)) {
            alert("Harap isi semua data dengan benar.");
            return;
        }

        produkList.push({ nama, stok, harga });
        resetForm();
    }

```

```

    tampilkanProduk();
}

function resetForm() {
    document.getElementById("namaProduk").value = "";
    document.getElementById("stokProduk").value = "";
    document.getElementById("hargaProduk").value = "";
}

function tampilkanProduk() {
    const tbody = document.getElementById("tableBody");
    tbody.innerHTML = "";

    let totalStok = 0;
    let totalHarga = 0;

    produkList.forEach((produk, index) => {
        const total = produk.stok * produk.harga;
        totalStok += produk.stok;
        totalHarga += total;

        const row = `
            <tr>
                <td>${index + 1}</td>
                <td>${produk.nama}</td>
                <td>${produk.stok}</td>
                <td>Rp${produk.harga.toLocaleString()}</td>
                <td>Rp${total.toLocaleString()}</td>
                <td>
                    <button class="w3-button w3-blue w3-small" onclick="editProduk(${index})">Edit</button>
                    <button class="w3-button w3-red w3-small"
onclick="hapusProduk(${index})">Hapus</button>
                </td>
            </tr>
        `;
        tbody.innerHTML += row;
    });

    document.getElementById("totalStok").innerText = totalStok;
    document.getElementById("totalHarga").innerText = "Rp" + totalHarga.toLocaleString();
}

function hapusProduk(index) {
    if (confirm("Yakin ingin menghapus produk ini?")) {
        produkList.splice(index, 1);
        tampilkanProduk();
    }
}

```

```

    }

    function editProduk(index) {
        const produk = produkList[index];
        const namaBaru = prompt("Edit nama produk:", produk.nama);
        const stokBaru = parseInt(prompt("Edit stok:", produk.stok));
        const hargaBaru = parseInt(prompt("Edit harga:", produk.harga));

        if (namaBaru && !isNaN(stokBaru) && !isNaN(hargaBaru)) {
            produkList[index] = { nama: namaBaru, stok: stokBaru, harga: hargaBaru };
            tampilkanProduk();
        }
    }

    function cariProduk() {
        const input = document.getElementById("searchInput").value.toLowerCase();
        const rows = document.querySelectorAll("#produkTable tbody tr");

        rows.forEach(row => {
            const nama = row.children[1].textContent.toLowerCase();
            row.style.display = nama.includes(input) ? "" : "none";
        });
    }
}
</script>

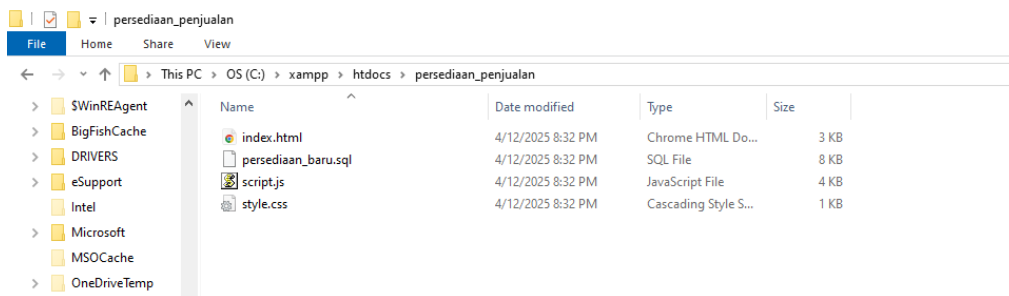
</body>
</html>

```

- **phpMyAdmin**

http://localhost/persediaan_penjualan/





- Index.html

```
<!DOCTYPE html>
<html lang="id">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Persediaan & Penjualan</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
```

```
  <!-- Logo Kucing -->
  <div class="logo-container">
    
  </div>
```

```
<h1>Manajemen Persediaan & Penjualan</h1>
```

```
<!-- Tabel Produk -->
<h2>Daftar Produk</h2>
<table>
  <thead>
    <tr>
      <th>ID</th>
      <th>Nama Produk</th>
      <th>Stok</th>
      <th>Harga</th>
      <th>Aksi</th>
    </tr>
  </thead>
  <tbody id="product-list">
    <!-- Data akan diisi oleh JavaScript -->
  </tbody>
</table>
```

```
<!-- Formulir Tambah Produk -->
<h2>Tambah Produk</h2>
```

```

<form id="add-product-form">
  <input type="text" id="product-name" placeholder="Nama Produk" required>
  <input type="number" id="product-stock" placeholder="Stok" required>
  <input type="number" id="product-price" placeholder="Harga" required>
  <button type="submit">Tambah Produk</button>
</form>

<!-- Formulir Penjualan -->
<h2>Tambah Penjualan</h2>
<form id="sell-product-form">
  <select id="sell-product">
    <!-- Opsi produk akan diisi oleh JavaScript -->
  </select>
  <input type="number" id="sell-quantity" placeholder="Jumlah" required>
  <button type="submit">Jual</button>
</form>

<!-- Riwayat Transaksi -->
<h2>Riwayat Transaksi</h2>
<table>
  <thead>
    <tr>
      <th>Nama Produk</th>
      <th>Jumlah</th>
      <th>Total Harga</th>
    </tr>
  </thead>
  <tbody id="transaction-list">
    <!-- Data akan diisi oleh JavaScript -->
  </tbody>
</table>

<script src="script.js"></script>
</body>
</html>

```

- Persediaan_baru.sql

```

-- phpMyAdmin SQL Dump
-- version 5.2.1
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Waktu pembuatan: 26 Feb 2025 pada 02.40
-- Versi server: 10.4.28-MariaDB
-- Versi PHP: 8.2.4

```

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
```

```
START TRANSACTION;
SET time_zone = "+00:00";
```

```
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;
```

```
--
-- Database: `persediaan_baru`
--
```

```
-----
```

```
--
-- Struktur dari tabel `barang`
--
```

```
CREATE TABLE `barang` (
  `id_barang` int(11) NOT NULL,
  `kode_barang` varchar(20) NOT NULL,
  `nama_barang` varchar(100) NOT NULL,
  `kategori_id` int(11) DEFAULT NULL,
  `stok` int(11) NOT NULL DEFAULT 0,
  `satuan` varchar(50) NOT NULL,
  `harga_beli` decimal(15,2) NOT NULL,
  `harga_jual` decimal(15,2) NOT NULL,
  `tgl_ditambahkan` datetime DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
-----
```

```
--
-- Struktur dari tabel `detail_pembelian`
--
```

```
CREATE TABLE `detail_pembelian` (
  `id_detail` int(11) NOT NULL,
  `id_pembelian` int(11) DEFAULT NULL,
  `id_barang` int(11) DEFAULT NULL,
  `jumlah` int(11) NOT NULL,
  `harga_satuan` decimal(15,2) NOT NULL,
  `subtotal` decimal(15,2) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
-----
```



```
--  
-- Struktur dari tabel `kategori`  
--
```

```
CREATE TABLE `kategori` (  
  `id_kategori` int(11) NOT NULL,  
  `nama_kategori` varchar(100) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
-- -----
```

```
--  
-- Struktur dari tabel `pemasok`  
--
```

```
CREATE TABLE `pemasok` (  
  `id_pemasok` int(11) NOT NULL,  
  `nama_pemasok` varchar(100) NOT NULL,  
  `kontak` varchar(50) DEFAULT NULL,  
  `alamat` text DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
-- -----
```

```
--  
-- Struktur dari tabel `pembelian`  
--
```

```
CREATE TABLE `pembelian` (  
  `id_pembelian` int(11) NOT NULL,  
  `kode_pembelian` varchar(20) NOT NULL,  
  `id_pemasok` int(11) DEFAULT NULL,  
  `tgl_pembelian` datetime DEFAULT current_timestamp(),  
  `total_harga` decimal(15,2) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
-- -----
```

```
--  
-- Struktur dari tabel `penjualan`  
--
```

```
CREATE TABLE `penjualan` (  
  `id_penjualan` int(11) NOT NULL,  
  `kode_penjualan` varchar(20) NOT NULL,  
  `tgl_penjualan` datetime DEFAULT current_timestamp(),
```

```
`total_harga` decimal(15,2) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
-- -----
```

```
--
```

```
-- Struktur dari tabel `stok_keluar`
```

```
--
```

```
CREATE TABLE `stok_keluar` (
  `id_stok_keluar` int(11) NOT NULL,
  `id_barang` int(11) DEFAULT NULL,
  `jumlah` int(11) NOT NULL,
  `tgl_keluar` datetime DEFAULT current_timestamp(),
  `keterangan` text DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
-- -----
```

```
--
```

```
-- Struktur dari tabel `stok_masuk`
```

```
--
```

```
CREATE TABLE `stok_masuk` (
  `id_stok_masuk` int(11) NOT NULL,
  `id_barang` int(11) DEFAULT NULL,
  `jumlah` int(11) NOT NULL,
  `tgl_masuk` datetime DEFAULT current_timestamp(),
  `keterangan` text DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
--
```

```
-- Indexes for dumped tables
```

```
--
```

```
--
```

```
-- Indeks untuk tabel `barang`
```

```
--
```

```
ALTER TABLE `barang`
  ADD PRIMARY KEY (`id_barang`),
  ADD UNIQUE KEY `kode_barang` (`kode_barang`),
  ADD KEY `kategori_id` (`kategori_id`);
```

```
--
```

```
-- Indeks untuk tabel `detail_pembelian`
```

```
--
```

```
ALTER TABLE `detail_pembelian`
```

```

ADD PRIMARY KEY (`id_detail`),
ADD KEY `fk_detail_pembelian` (`id_pembelian`),
ADD KEY `fk_detail_pembelian_barang` (`id_barang`);

--
-- Indeks untuk tabel `kategori`
--
ALTER TABLE `kategori`
  ADD PRIMARY KEY (`id_kategori`);

--
-- Indeks untuk tabel `pemasok`
--
ALTER TABLE `pemasok`
  ADD PRIMARY KEY (`id_pemasok`);

--
-- Indeks untuk tabel `pembelian`
--
ALTER TABLE `pembelian`
  ADD PRIMARY KEY (`id_pembelian`),
  ADD UNIQUE KEY `kode_pembelian` (`kode_pembelian`),
  ADD KEY `fk_pembelian_pemasok` (`id_pemasok`);

--
-- Indeks untuk tabel `penjualan`
--
ALTER TABLE `penjualan`
  ADD PRIMARY KEY (`id_penjualan`),
  ADD UNIQUE KEY `kode_penjualan` (`kode_penjualan`);

--
-- Indeks untuk tabel `stok_keluar`
--
ALTER TABLE `stok_keluar`
  ADD PRIMARY KEY (`id_stok_keluar`),
  ADD KEY `fk_stok_keluar_barang` (`id_barang`);

--
-- Indeks untuk tabel `stok_masuk`
--
ALTER TABLE `stok_masuk`
  ADD PRIMARY KEY (`id_stok_masuk`),
  ADD KEY `fk_stok_masuk_barang` (`id_barang`);

--
-- AUTO_INCREMENT untuk tabel yang dibuang

```

```
--  
  
--  
-- AUTO_INCREMENT untuk tabel `barang`  
--  
ALTER TABLE `barang`  
  MODIFY `id_barang` int(11) NOT NULL AUTO_INCREMENT;  
  
--  
-- AUTO_INCREMENT untuk tabel `detail_pembelian`  
--  
ALTER TABLE `detail_pembelian`  
  MODIFY `id_detail` int(11) NOT NULL AUTO_INCREMENT;  
  
--  
-- AUTO_INCREMENT untuk tabel `kategori`  
--  
ALTER TABLE `kategori`  
  MODIFY `id_kategori` int(11) NOT NULL AUTO_INCREMENT;  
  
--  
-- AUTO_INCREMENT untuk tabel `pemasok`  
--  
ALTER TABLE `pemasok`  
  MODIFY `id_pemasok` int(11) NOT NULL AUTO_INCREMENT;  
  
--  
-- AUTO_INCREMENT untuk tabel `pembelian`  
--  
ALTER TABLE `pembelian`  
  MODIFY `id_pembelian` int(11) NOT NULL AUTO_INCREMENT;  
  
--  
-- AUTO_INCREMENT untuk tabel `penjualan`  
--  
ALTER TABLE `penjualan`  
  MODIFY `id_penjualan` int(11) NOT NULL AUTO_INCREMENT;  
  
--  
-- AUTO_INCREMENT untuk tabel `stok_keluar`  
--  
ALTER TABLE `stok_keluar`  
  MODIFY `id_stok_keluar` int(11) NOT NULL AUTO_INCREMENT;  
  
--  
-- AUTO_INCREMENT untuk tabel `stok_masuk`  
--
```

```

ALTER TABLE `stok_masuk`
  MODIFY `id_stok_masuk` int(11) NOT NULL AUTO_INCREMENT;

--
-- Ketidakeleluasaan untuk tabel pelimpahan (Dumped Tables)
--

--
-- Ketidakeleluasaan untuk tabel `barang`
--
ALTER TABLE `barang`
  ADD CONSTRAINT `barang_ibfk_1` FOREIGN KEY (`kategori_id`) REFERENCES `kategori`
(`id_kategori`) ON DELETE SET NULL ON UPDATE CASCADE;

--
-- Ketidakeleluasaan untuk tabel `detail_pembelian`
--
ALTER TABLE `detail_pembelian`
  ADD CONSTRAINT `fk_detail_pembelian` FOREIGN KEY (`id_pembelian`) REFERENCES
`pembelian` (`id_pembelian`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `fk_detail_pembelian_barang` FOREIGN KEY (`id_barang`) REFERENCES
`barang` (`id_barang`) ON DELETE CASCADE ON UPDATE CASCADE;

--
-- Ketidakeleluasaan untuk tabel `pembelian`
--
ALTER TABLE `pembelian`
  ADD CONSTRAINT `fk_pembelian_pemasok` FOREIGN KEY (`id_pemasok`) REFERENCES
`pemasok` (`id_pemasok`) ON DELETE SET NULL ON UPDATE CASCADE;

--
-- Ketidakeleluasaan untuk tabel `stok_keluar`
--
ALTER TABLE `stok_keluar`
  ADD CONSTRAINT `fk_stok_keluar_barang` FOREIGN KEY (`id_barang`) REFERENCES `barang`
(`id_barang`) ON DELETE CASCADE ON UPDATE CASCADE;

--
-- Ketidakeleluasaan untuk tabel `stok_masuk`
--
ALTER TABLE `stok_masuk`
  ADD CONSTRAINT `fk_stok_masuk_barang` FOREIGN KEY (`id_barang`) REFERENCES `barang`
(`id_barang`) ON DELETE CASCADE ON UPDATE CASCADE;
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;

```

```
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

- Style.css

```
body {  
    font-family: Arial, sans-serif;  
    text-align: center;  
    background-color: #f4f4f4;  
}  
  
.logo-container {  
    text-align: center;  
    margin-top: 20px;  
}  
  
.logo {  
    width: 120px; /* Sesuaikan ukuran logo */  
    height: auto;  
    display: block;  
    margin: 0 auto;  
}  
  
h1, h2 {  
    color: #333;  
}  
  
table {  
    width: 80%;  
    margin: 20px auto;  
    border-collapse: collapse;  
    background: #fff;  
}  
  
th, td {  
    border: 1px solid #ddd;  
    padding: 10px;  
    text-align: center;  
}  
  
th {  
    background-color: #4CAF50;  
    color: white;  
}  
  
form {  
    margin: 20px auto;  
    width: 80%;  
    padding: 10px;
```

```
background: white;
border-radius: 5px;
}
```

```
input, select, button {
  margin: 5px;
  padding: 8px;
}
```

- Script.js

```
document.addEventListener("DOMContentLoaded", () => {
  const productList = document.getElementById("product-list");
  const transactionList = document.getElementById("transaction-list");
  const sellProductSelect = document.getElementById("sell-product");

  let products = [];
  let transactions = [];

  // Tambah Produk
  document.getElementById("add-product-form").addEventListener("submit", (e) => {
    e.preventDefault();

    const name = document.getElementById("product-name").value;
    const stock = parseInt(document.getElementById("product-stock").value);
    const price = parseInt(document.getElementById("product-price").value);

    if (name && stock > 0 && price > 0) {
      const newProduct = { id: products.length + 1, name, stock, price };
      products.push(newProduct);
      updateProductTable();
      updateProductDropdown();
    }
  });

  // Jual Produk
  document.getElementById("sell-product-form").addEventListener("submit", (e) => {
    e.preventDefault();

    const productId = parseInt(sellProductSelect.value);
    const quantity = parseInt(document.getElementById("sell-quantity").value);

    const product = products.find(p => p.id === productId);
    if (product && quantity > 0 && quantity <= product.stock) {
      product.stock -= quantity;
    }
  });
});
```

```

        transactions.push({ name: product.name, quantity, total: quantity * product.price });
        updateProductTable();
        updateTransactionTable();
    } else {
        alert("Stok tidak mencukupi!");
    }
});

```

// Perbarui Tabel Produk

```

function updateProductTable() {
    productList.innerHTML = "";
    products.forEach(product => {
        const row = `<tr>
            <td>${product.id}</td>
            <td>${product.name}</td>
            <td>${product.stock}</td>
            <td>${product.price}</td>
            <td><button onclick="deleteProduct(${product.id})">Hapus</button></td>
        </tr>`;
        productList.innerHTML += row;
    });
}

```

// Perbarui Dropdown Produk

```

function updateProductDropdown() {
    sellProductSelect.innerHTML = "";
    products.forEach(product => {
        sellProductSelect.innerHTML += `<option
value="${product.id}">${product.name}</option>`;
    });
}

```

// Perbarui Tabel Transaksi

```

function updateTransactionTable() {
    transactionList.innerHTML = "";
    transactions.forEach(t => {
        const row = `<tr>
            <td>${t.name}</td>
            <td>${t.quantity}</td>
            <td>${t.total}</td>
        </tr>`;
        transactionList.innerHTML += row;
    });
}

```



```
}
```

```
// Hapus Produk
```

```
window.deleteProduct = function(id) {  
  products = products.filter(p => p.id !== id);  
  updateProductTable();  
  updateProductDropdown();  
};  
});
```