

Pertemuan 6 CRUD

Program SQL Database

1. Membuat Database

```
CREATE DATABASE IF NOT EXISTS latihan_crud;  
USE latihan_crud;
```

- **CREATE DATABASE IF NOT EXISTS latihan_crud;**
 - Membuat database baru dengan nama **latihan_crud**.
 - **IF NOT EXISTS** artinya: kalau sudah ada, jangan error, cukup lewati.
- **USE latihan_crud;**
 - Memilih database **latihan_crud** sebagai database aktif.
 - Semua perintah SQL berikutnya akan dijalankan di database ini.

2. Membuat Tabel

```
CREATE TABLE IF NOT EXISTS barang (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  nama VARCHAR(100),  
  harga INT  
) ENGINE=InnoDB;
```

- **CREATE TABLE IF NOT EXISTS barang**
 - Membuat tabel baru dengan nama **barang**.
 - **IF NOT EXISTS** = kalau tabel **barang** sudah ada, MySQL tidak error.
- **id INT AUTO_INCREMENT PRIMARY KEY**
 - Kolom **id** bertipe **integer**.

- `AUTO_INCREMENT` → nilainya otomatis bertambah setiap kali menambah data baru.
 - `PRIMARY KEY` → menjadi kunci unik untuk tiap baris.
 - `nama VARCHAR(100)`
 - Kolom `nama` bertipe **varchar** (teks pendek) maksimal 100 karakter.
 - Digunakan untuk menyimpan nama barang.
 - `harga INT`
 - Kolom `harga` bertipe **integer** (bilangan bulat).
 - Digunakan untuk menyimpan harga barang.
 - `) ENGINE=InnoDB;`
 - Menentukan mesin database yang dipakai adalah **InnoDB**.
 - InnoDB mendukung transaksi, foreign key, dan lebih stabil untuk aplikasi modern.
-

3. Alur sederhananya

1. Pertama kamu buat database bernama `latihan_crud`.
 2. Di dalamnya, dibuat tabel `barang`.
 3. Tabel ini punya 3 kolom:
 - `id` (nomor otomatis)
 - `nama` (teks nama barang)
 - `harga` (angka harga barang)
 4. Tabel siap dipakai oleh PHP CRUD kamu untuk **tambah, lihat, ubah, hapus barang**.
-

👉 Jadi file `produk.sql` ini adalah **skema awal** untuk latihan CRUD. Tanpa menjalankan file ini di MySQL/phpMyAdmin, tabel `barang` tidak akan ada, sehingga `index.php`, `tambah.php`, `edit.php`, dan `hapus.php` tidak bisa berjalan.

Materi Bootstrap

1. Apa itu Bootstrap?

- **Bootstrap** adalah framework **CSS + JavaScript** yang sudah berisi kumpulan style, komponen, dan layout siap pakai.
 - Fungsinya: mempercepat membuat tampilan website yang **rapi, modern, dan responsif** tanpa harus menulis CSS dari nol.
 - Versi terbaru sekarang adalah **Bootstrap 5** (yang kamu pakai di header.php lewat CDN).
-

2. Kenapa Bootstrap dipakai?

- ✓ **Responsif otomatis** → tampilan menyesuaikan layar HP, tablet, laptop.
 - ✓ **Komponen siap pakai** → tombol, navbar, form, tabel, alert, modal, dll.
 - ✓ **Mudah dipelajari** → cukup pakai class di elemen HTML.
 - ✓ **Konsisten** → hasil desain seragam, tidak perlu atur ulang style tiap halaman.
-

3. Cara Memakai Bootstrap

Ada dua cara:

1. **CDN** (paling mudah) – cukup tambahkan link CSS & JS dari internet:

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
```

2. **Download file** – unduh dan simpan CSS/JS Bootstrap di folder project (butuh setting manual).

Di CRUD kamu, kamu sudah pakai **CDN**.

4. Grid System (Layout Utama Bootstrap)

- Bootstrap pakai sistem **grid 12 kolom**.
- Kamu atur lebar konten dengan class seperti `col-`, `col-md-`, `col-lg-`.

Contoh:

```
<div class="container">
  <div class="row">
    <div class="col-md-8">Konten utama (8 kolom)</div>
    <div class="col-md-4">Sidebar (4 kolom)</div>
  </div>
</div>
```

➡ Di layar besar, konten terbagi 8 + 4 kolom. Di layar kecil, otomatis turun ke bawah (responsif).

5. Komponen Populer Bootstrap

1. Button

```
<button class="btn btn-primary">Primary</button>
<button class="btn btn-success">Success</button>
<button class="btn btn-danger">Delete</button>
```

→ Warna & style tombol langsung jadi.

2. Form

```
<input type="text" class="form-control" placeholder="Nama">
```

→ Input jadi rapi dan full width.

3. Tabel

```
<table class="table table-striped table-bordered">
  <tr><th>No</th><th>Nama</th><th>Harga</th></tr>
  <tr><td>1</td><td>Apel</td><td>10000</td></tr>
</table>
```

→ Tabel ada garis, zebra strip, dan responsif.

4. Alert

```
<div class="alert alert-success">Berhasil!</div>
<div class="alert alert-danger">Error!</div>
```

→ Pesan notifikasi berwarna otomatis.

5. Navbar

```
<nav class="navbar navbar-dark bg-dark">
  <div class="container">
    <a class="navbar-brand" href="#">MyApp</a>
  </div>
</nav>
```

→ Navbar gelap siap pakai.

6. Utilities (Class Bantuan Cepat)

Bootstrap punya class-class kecil untuk styling instan:

- **Margin & padding:** `m-3` (margin 1rem), `p-2` (padding 0.5rem).
- **Text:** `text-center`, `fw-bold`, `text-muted`.
- **Background:** `bg-light`, `bg-dark`, `bg-primary`.
- **Flexbox:** `d-flex`, `justify-content-between`, `align-items-center`.

Contoh:

```
<div class="d-flex justify-content-between bg-light p-3">
  <span>Kiri</span>
  <span>Kanan</span>
</div>
```

7. Hubungan dengan CRUD kamu

- **Navbar** di `header.php` pakai `navbar navbar-dark bg-dark`.
- **Card** dipakai untuk form tambah/edit.
- **Tabel** pakai `table table-striped table-bordered align-middle`.
- **Alert** dipakai untuk pesan sukses/error.

Jadi CRUD kamu sudah menerapkan **komponen inti Bootstrap**: navbar, card, form, tabel, alert.

Kesimpulan:

- Bootstrap adalah "perpustakaan CSS/JS siap pakai" untuk bikin website cepat, rapi, dan responsif.
- Cukup tambahkan class Bootstrap di elemen HTML.
- CRUD kamu sudah memanfaatkan bagian penting: navbar, form, tabel, alert, card.

Program Koneksi PHP

Betul 

kode `koneksi.php` kamu itu sudah **benar** dan cukup untuk latihan CRUD sederhana dengan `mysqli`.

Agar lebih jelas, aku kasih beberapa catatan kecil supaya kamu lebih paham fungsinya:

```

<?php
// Membuat koneksi ke MySQL
$koneksi = mysqli_connect("localhost", "root", "", "latihan_crud");

// Cek apakah koneksi berhasil
if (!$koneksi) {
    die("Koneksi gagal: " . mysqli_connect_error());
}

// echo "Koneksi berhasil!"; // (opsional) bisa dipakai untuk tes
?>

```

Penjelasan baris per baris:

1. `mysqli_connect("localhost", "root", "", "latihan_crud");`
 - `"localhost"` → alamat server database (default: localhost / 127.0.0.1).
 - `"root"` → username default MySQL di XAMPP/Laragon.
 - `""` → password default (kosong). Kalau MySQL-mu ada password, isi di sini.
 - `"latihan_crud"` → nama database yang sudah kamu buat.
2. `if (!$koneksi)`
Mengecek apakah koneksi gagal. Jika iya, tampilkan error.
3. `die("Koneksi gagal: " . mysqli_connect_error());`
Menghentikan program dan menampilkan pesan error jika koneksi tidak berhasil.

💡 Tips:

- Simpan file ini di folder project kamu dengan nama `koneksi.php`.
- Di file lain tinggal tambahkan:

```
<?php include "koneksi.php"; ?>
```

supaya otomatis terkoneksi.

Program Header PHP

1. Deklarasi HTML & Head

```
<html lang="id">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Latihan CRUD</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
```

- `<html lang="id">` → memberi tahu browser bahwa bahasa utama dokumen ini **Bahasa Indonesia**.
- `<meta charset="utf-8">` → agar teks bisa tampil dengan benar (mendukung semua karakter).
- `<meta name="viewport"...">` → supaya tampilan **responsif** (menyesuaikan layar HP atau PC).
- `<title>Latihan CRUD</title>` → teks judul di tab browser.
- `<link ... bootstrap.min.css>` → memuat **Bootstrap 5** via CDN, agar bisa pakai class siap pakai (`btn` , `navbar` , `container` , dll).

2. Body dan Background

```
<body class="bg-light">
```

- Membuka body dokumen.

- `class="bg-light"` memberi background warna abu-abu muda (warna default Bootstrap).

3. Navbar (Menu Atas)

```
<nav class="navbar navbar-dark bg-dark">
<div class="container">
<a class="navbar-brand" href="index.php">Latihan CRUD</a>
<div class="d-flex gap-2">
<a href="index.php" class="btn btn-outline-light btn-sm">Daftar</a>
<a href="tambah.php" class="btn btn-success btn-sm">+ Tambah</a>
</div>
</div>
</nav>
```

- `<nav class="navbar navbar-dark bg-dark">` → membuat navbar berwarna gelap.
- `<div class="container">` → agar isi navbar rata tengah (tidak mepet pinggir).
- `` → logo/tulisan brand → "Latihan CRUD".
- `<div class="d-flex gap-2">` → menaruh dua tombol di kanan navbar dengan jarak kecil.
 - Tombol **Daftar** (`btn btn-outline-light btn-sm`) → warna border putih kecil.
 - Tombol **+ Tambah** (`btn btn-success btn-sm`) → tombol hijau kecil.

4. Bagian Utama Halaman

```
<main class="container py-4">
```

- `<main>` → elemen semantik HTML untuk isi utama.
- `class="container py-4"` → memberi ruang (padding `py-4`) supaya isi halaman tidak terlalu mepet ke atas.

Nantinya isi konten (tabel daftar, form tambah/edit, dll) akan diletakkan di dalam `<main> ... </main>`.

👉 Jadi singkatnya:

`header.php` ini berfungsi untuk **menyiapkan struktur awal halaman + navbar Bootstrap**. Supaya rapi, di tiap halaman CRUD kamu cukup `include "header.php";` di atas, lalu nanti tutup dengan `include "footer.php";`.

Program Footer PHP

```
</main>
```

➡ Menutup tag `<main>` yang sebelumnya dibuka di `header.php`. Jadi semua konten utama (tabel, form, dll) harus ada di dalam `<main> ... </main>`.

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
```

➡ Memuat file **JavaScript Bootstrap** (via CDN).

- Fungsinya untuk membuat komponen Bootstrap yang butuh JavaScript bisa jalan (misalnya modal, dropdown, toast, collapse).
- `bundle.min.js` sudah termasuk **Popper.js**, jadi cukup sekali load.

```
</body>  
</html>
```

➡ Menutup dokumen HTML.

📌 Jadi struktur file kamu akan seperti ini:

- **header.php** → mulai dari `<html>` sampai `<main>`
- **footer.php** → menutup `</main>` sampai `</html>`

Contoh penggunaan di halaman utama (`index.php`):

```
<?php include "header.php"; ?>

<h1>Halo, ini halaman index</h1>
<p>Isi tabel atau form CRUD diletakkan di sini.</p>

<?php include "footer.php"; ?>
```

Dengan pola ini, semua halaman akan punya **header + navbar** yang sama dan **penutup yang sama**.

Program Index (Read) PHP

1. Include Koneksi & Header

```
<?php include "koneksi.php";
include "header.php";
$no = 1;
if (isset($_GET['msg'])): ?>
```

- `include "koneksi.php";` → menghubungkan PHP dengan database MySQL (supaya bisa ambil data).
- `include "header.php";` → menampilkan bagian header + navbar Bootstrap.
- `$no = 1;` → variabel nomor untuk penomoran tabel (kolom **No.**). Akan bertambah otomatis dengan `$no++`.
- `if (isset($_GET['msg'])):` → kalau ada pesan dari URL (misalnya `?msg=Data berhasil dihapus`), tampilkan alert.

2. Alert Pesan

```
<div class="alert alert-info"><?= htmlspecialchars($_GET['msg']) ?></div>
```

- Menampilkan pesan info di kotak biru Bootstrap (`alert-info`).
- `htmlspecialchars()` dipakai supaya teks pesan aman (mencegah XSS).

Contoh: setelah hapus data, halaman diarahkan kembali ke index.php dengan `?msg=Data berhasil dihapus` .

3. Membuat Card Bootstrap

```
<div class="card shadow-sm">
<div class="card-header fw-semibold">Daftar Barang</div>
<div class="card-body">
```

- `card` → kotak Bootstrap rapi.
- `shadow-sm` → efek bayangan tipis.
- `card-header` → bagian atas card, ditulis **Daftar Barang**.
- `fw-semibold` → teks agak tebal.
- `card-body` → isi card (tabel).

4. Membuat Tabel

```
<div class="table-responsive">
<table class="table table-striped table-bordered align-middle">
<thead class="table-dark">
<tr>
<th style="width:60px">No.</th>
<th style="width:70px">ID</th>
<th>Nama</th>
<th style="width:160px">Harga</th>
<th style="width:160px">Aksi</th>
</tr>
</thead>
<tbody>
```

- `table-responsive` → tabel bisa digeser di layar kecil.
- `table table-striped table-bordered align-middle` → tabel Bootstrap dengan garis, baris bergaris, dan teks tengah vertikal.
- `thead class="table-dark"` → header tabel berwarna hitam.
- Kolom tabel: **No. | ID | Nama | Harga | Aksi.**

5. Menampilkan Data dari Database

```
<?php
$result = mysqli_query($koneksi, "SELECT * FROM barang ORDER BY id DESC
C");
while ($row = mysqli_fetch_assoc($result)):
?>
<tr>
<td><?= $no++ ?></td>
<td>AX<?= $row['id'] ?></td>
<td><?= htmlspecialchars($row['nama']) ?></td>
<td>Rp <?= number_format((int)$row['harga'], 0, ',', '.') ?></td>
<td>
<a href="edit.php?id=<?= $row['id'] ?>" class="btn btn-warning btn-sm">Edi
t</a>
<a href="hapus.php?id=<?= $row['id'] ?>" class="btn btn-danger btn-sm" on
click="return confirm('Hapus data ini?')">Hapus</a>
</td>
</tr>
<?php endwhile; ?>
```

- `mysqli_query(... "SELECT * FROM barang ORDER BY id DESC")` → ambil semua data dari tabel `barang`,urut dari ID terbesar (data terbaru di atas).
- `while ($row = mysqli_fetch_assoc($result)):` → looping tiap baris hasil query.
- `<td><?= $no++ ?></td>` → tampilkan nomor urut (mulai dari 1).

- `<td>AX<?= $row['id'] ?></td>` → menampilkan ID barang dengan prefix **AX** (contoh: `AX1`, `AX2`).
- `<td><?= htmlspecialchars($row['nama']) ?></td>` → menampilkan nama barang (aman dari script).
- `<td>Rp <?= number_format((int)$row['harga'], 0, ',', '.') ?></td>` → menampilkan harga dalam format rupiah (misalnya `Rp 10.000`).
- Kolom aksi → tombol **Edit** (kuning) & **Hapus** (merah).
 - Tombol hapus punya `onclick="return confirm('Hapus data ini?')"` → muncul konfirmasi sebelum hapus.

6. Tutup Card & Footer

```
</tbody>
</table>
</div>
</div>
</div>
```

```
<?php include "footer.php"; ?>
```

- Menutup tabel, card, dan container.
- `include "footer.php";` → menambahkan bagian bawah HTML (`</main> ... </html>` + script Bootstrap).

✨ Kesimpulan:

- `index.php` adalah halaman utama untuk **Read (lihat data)**.
- Menggunakan Bootstrap card & table agar tampilan rapi.
- Ada nomor otomatis, format harga rupiah, dan pesan alert.
- Aksi Edit & Hapus langsung terhubung ke `edit.php` dan `hapus.php`.

Program Tambah PHP

1. Include koneksi & header

```
<?php include "koneksi.php"; ?>
<?php include "header.php"; ?>
```

- `koneksi.php` → supaya bisa terhubung dengan database.
- `header.php` → memanggil bagian atas halaman (navbar + pembuka `<main>`).

2. Persiapan variabel & validasi

```
$errors = [];
$nama = '';
$harga = '';
```

- `$errors` → array untuk menyimpan pesan error validasi.
- `$nama` dan `$harga` → variabel untuk menyimpan input dari form.

3. Logika saat tombol simpan ditekan

```
if (isset($_POST['simpan'])) {
    $nama = trim($_POST['nama'] ?? '');
    $harga = trim($_POST['harga'] ?? '');
    if ($nama === '') $errors[] = 'Nama wajib diisi';
    if ($harga === '' || (int)$harga < 0) $errors[] = 'Harga harus diisi dan tidak negatif';

    if (!$errors) {
        mysqli_query($koneksi, "INSERT INTO barang (nama, harga)
        VALUES ('".mysqli_real_escape_string($koneksi,$nama)."',".(int)$harga.
        a.'')");
        header('Location: index.php?msg=Data berhasil ditambah');
```

```
    exit;
  }
}
```

- `if (isset($_POST['simpan']))` → cek apakah form sudah dikirim.
- `trim()` → menghapus spasi berlebih di awal/akhir input.
- Validasi:
 - Jika nama kosong → error.
 - Jika harga kosong atau negatif → error.
- Kalau **tidak ada error**:
 - Simpan data ke tabel `barang` dengan `INSERT`.
 - `mysqli_real_escape_string` → mencegah SQL Injection.
 - `(int)$harga` → memastikan harga angka.
 - Setelah simpan, diarahkan balik ke `index.php` dengan pesan sukses.

4. Tampilan Form Tambah

```
<div class="row justify-content-center">
  <div class="col-md-6">
    <div class="card shadow-sm">
      <div class="card-header fw-semibold">Tambah Barang</div>
      <div class="card-body">
```

- Gunakan **Bootstrap grid** → form ditaruh di tengah (`justify-content-center`) dengan lebar 6 kolom (`col-md-6`).
- `card shadow-sm` → kotak form dengan bayangan.
- `card-header` → judul form "Tambah Barang".

5. Menampilkan Error (jika ada)


```

<?php if ($errors): ?>
<div class="alert alert-danger">
  <ul class="mb-0">
    <?php foreach ($errors as $e): ?><li><?= htmlspecialchars($e) ?></li><?ph
p endforeach; ?>
  </ul>
</div>
<?php endif; ?>

```

- Jika ada error, tampilkan di kotak merah (`alert-danger`).
- Semua pesan error ditampilkan sebagai `` .

6. Form Input

```

<form method="post" novalidate>
  <div class="mb-3">
    <label class="form-label">Nama</label>
    <input type="text" name="nama" class="form-control" value="<?= htmlspecialchars($nama) ?>" required>
  </div>
  <div class="mb-3">
    <label class="form-label">Harga (Rp)</label>
    <input type="number" name="harga" class="form-control" value="<?= htmlspecialchars($harga) ?>" required>
  </div>
  <div class="d-flex gap-2">
    <button type="submit" name="simpan" class="btn btn-primary">Simpan</button>
    <a href="index.php" class="btn btn-secondary">Batal</a>
  </div>
</form>

```

- `method="post"` → kirim data dengan POST (lebih aman daripada GET).

- Input "Nama" dan "Harga":
 - `form-control` → style Bootstrap.
 - `value="<?= htmlspecialchars($nama) ?>"` → supaya kalau ada error, data yang sudah diketik tidak hilang.
- Tombol **Simpan** (submit) & **Batal** (kembali ke index).

7. Tutup & Footer

```
<?php include "footer.php"; ?>
```

- Menutup halaman dengan `footer.php` (script Bootstrap + tag penutup HTML).

✨ Kesimpulan:

- `tambah.php` adalah implementasi **Create** dalam CRUD.
- Ada **validasi sederhana** (nama tidak boleh kosong, harga harus angka ≥ 0).
- Jika sukses → data masuk ke tabel `barang` → balik ke index dengan pesan sukses.
- Kalau gagal → tampilkan error merah di atas form.

Program Edit PHP

1) Include koneksi & header

```
<?php include "koneksi.php"; ?>
<?php include "header.php"; ?>
```

Supaya terhubung ke database dan memakai layout Bootstrap (`header.php` membuka `<main>`).

2) Ambil `id` lalu baca data lama

```

$id = (int)($_GET['id'] ?? 0);
$data = mysqli_query($koneksi, "SELECT * FROM barang WHERE id=$id");
$row = mysqli_fetch_assoc($data);
if (!$row) { echo '<div class="alert alert-danger">Data tidak ditemukan</div>'; include 'footer.php'; exit; }

```

- `$_GET['id']` di-cast ke **int** → mencegah injeksi via URL.
- Query data barang berdasar `id`.
- Jika tidak ada, tampilkan alert dan **berhenti** (biar tidak error di bawahnya).

3) Siapkan variabel & error state

```

$errors = [];
$nama = $row['nama'];
$harga = $row['harga'];

```

- Prefill variabel dengan data lama, supaya form terisi otomatis.

4) Proses submit (validasi & update)

```

if (isset($_POST['update'])) {
    $nama = trim($_POST['nama'] ?? '');
    $harga = trim($_POST['harga'] ?? '');
    if ($nama === '') $errors[] = 'Nama wajib diisi';
    if ($harga === '' || (int)$harga < 0) $errors[] = 'Harga harus diisi dan tidak negatif';

    if (!$errors) {
        mysqli_query(
            $koneksi,
            "UPDATE barang SET nama='".mysqli_real_escape_string($koneksi,$nama)."', harga="(int)$harga." WHERE id=$id"
        );
    }
}

```

```

header('Location: index.php?msg=Data berhasil diupdate');
exit;
}
}

```

- **Validasi sederhana:** nama wajib, harga angka ≥ 0 .
- **Keamanan input:**
 - `mysqli_real_escape_string` pada `nama` .
 - `(int)$harga` & `(int)$id` memastikan tipe integer.
- Jika valid → `UPDATE` → **redirect** ke `index.php` sambil membawa pesan sukses `msg=...` .

Catatan: untuk produksi, sebaiknya gunakan prepared statements (mysqli/PDO) agar lebih aman & rapi.

5) Tampilkan form (prefilled + pesan error)

```

<?php if ($errors): ?>
  <div class="alert alert-danger">
    <ul class="mb-0">
      <?php foreach ($errors as $e): ?><li><?= htmlspecialchars($e) ?></li><?p
hp endforeach; ?>
    </ul>
  </div>
<?php endif; ?>

```

- Jika ada error, tampilkan di alert merah.
- `htmlspecialchars()` mencegah XSS saat menampilkan teks error.

```

<input type="text" name="nama" class="form-control" value="<?= htmlspecialchars($nama) ?>" required>

```

```
<input type="number" name="harga" class="form-control" value="<?= htmlspecialchars($harga) ?>" required>
```

- Field sudah **terisi nilai lama**; kalau validasi gagal, nilai yang sudah diketik tidak hilang.

Tombol:

```
<button type="submit" name="update" class="btn btn-primary">Update</button>  
<a href="index.php" class="btn btn-secondary">Kembali</a>
```

6) Footer

```
<?php include "footer.php"; ?>
```

Menutup layout (memuat JS Bootstrap + `</main></body></html>`).

Alur ringkas (mindset Update)

1. Ambil `id` dari URL.
2. Ambil data lama → jika tidak ada, hentikan.
3. Tampilkan form dengan nilai lama (prefill).
4. Saat submit:
 - Validasi input
 - Jika lolos → `UPDATE`
 - Redirect kembali ke daftar dengan pesan sukses

Program Hapus PHP

Isi kode `hapus.php` kamu:

```
<?php
include "koneksi.php";
$id = (int)($_GET['id'] ?? 0);
if ($id > 0) {
    mysqli_query($koneksi, "DELETE FROM barang WHERE id=$id");
}
header('Location: index.php?msg=Data berhasil dihapus');
exit;
```

Penjelasan per baris

1. `include "koneksi.php";`

→ Menghubungkan file ini ke database MySQL. Jadi kita bisa menjalankan query SQL.

2. `$id = (int)($_GET['id'] ?? 0);`

→ Mengambil parameter `id` dari URL, contoh: `hapus.php?id=3`.

→ `(int)` artinya dipaksa jadi angka (supaya aman & tidak bisa disusupi teks).

→ Kalau tidak ada `id`, otomatis nilainya jadi `0`.

3. `if ($id > 0) { ... }`

→ Cek apakah `id` valid (lebih besar dari 0).

→ Kalau iya, baris di dalam `if` dijalankan.

4. `mysqli_query($koneksi, "DELETE FROM barang WHERE id=$id");`

→ Menjalankan perintah SQL untuk menghapus data barang berdasarkan `id`.

→ Contoh kalau `$id = 3`, maka query-nya:

```
DELETE FROM barang WHERE id=3
```

→ Jadi data dengan `id=3` di tabel `barang` akan hilang.

5. `header('Location: index.php?msg=Data berhasil dihapus');`

→ Setelah eksekusi selesai, pengguna langsung diarahkan (redirect) kembali ke `index.php`.

→ Ada tambahan `?msg=Data berhasil dihapus` di URL, supaya di `index.php` bisa tampil alert pesan sukses.

6. `exit;`

→ Menghentikan program agar tidak ada kode lain yang dijalankan setelah redirect.

Alur Singkatnya:

- User klik tombol **Hapus** → diarahkan ke `hapus.php?id=...`
- `hapus.php` ambil `id` dari URL → jalankan query DELETE → kembali ke `index.php`
- Di `index.php` tampil pesan: "Data berhasil dihapus".

Bootstrap 5 Cheat Sheet (CRUD Edition)

1. Layout & Container

- `container` → pembungkus dengan margin otomatis.
- `row` → baris untuk grid.
- `col`, `col-md-6`, `col-lg-4` → kolom grid (total 12).

 Contoh:

```
<div class="container">
  <div class="row">
    <div class="col-md-8">Kolom 8</div>
    <div class="col-md-4">Kolom 4</div>
  </div>
</div>
```

2. Typography (Teks)

- `fw-bold` , `fw-semibold` → tebal.
- `text-center` , `text-start` , `text-end` → rata teks.
- `text-muted` → abu-abu lembut.
- `h1` , `h2` , `h3` → ukuran heading bawaan Bootstrap.

📌 Contoh:

```
<h3 class="fw-bold text-center text-muted">Daftar Barang</h3>
```

3. Buttons

- `btn btn-primary` → tombol biru.
- `btn btn-success` → tombol hijau.
- `btn btn-danger` → tombol merah.
- `btn btn-warning` → tombol kuning.
- `btn-outline-light` → tombol outline putih.
- `btn-sm` → tombol kecil.

📌 Contoh:

```
<button class="btn btn-success btn-sm">+ Tambah</button>
```

4. Forms

- `form-control` → input rapi.
- `form-label` → label input.
- `mb-3` → jarak bawah antar form.

📌 Contoh:

```
<div class="mb-3">
  <label class="form-label">Nama</label>
```



```
<input type="text" class="form-control">
</div>
```

5. Cards (Kotak Konten)

- `card` → komponen dasar.
- `card-header` → judul card.
- `card-body` → isi card.
- `shadow-sm` → efek bayangan.

📌 Contoh:

```
<div class="card shadow-sm">
  <div class="card-header fw-semibold">Tambah Barang</div>
  <div class="card-body">Isi Form</div>
</div>
```

6. Tables

- `table` → tabel dasar.
- `table-striped` → zebra strip.
- `table-bordered` → ada garis.
- `align-middle` → teks rata tengah vertikal.
- `table-responsive` → tabel bisa digeser di HP.

📌 Contoh:

```
<div class="table-responsive">
  <table class="table table-striped table-bordered align-middle">
    <thead class="table-dark">
      <tr><th>No</th><th>Nama</th><th>Harga</th></tr>
    </thead>
```

```
</table>
</div>
```

7. Alerts (Pesan)

- `alert alert-success` → pesan hijau.
- `alert alert-danger` → pesan merah.
- `alert alert-info` → pesan biru.

 Contoh:

```
<div class="alert alert-success">Data berhasil ditambah</div>
```

8. Navbar

- `navbar navbar-dark bg-dark` → navbar gelap.
- `navbar-brand` → nama/logo aplikasi.
- `d-flex gap-2` → tombol kanan berjejer dengan jarak.

 Contoh:

```
<nav class="navbar navbar-dark bg-dark">
  <div class="container">
    <a class="navbar-brand" href="#">Latihan CRUD</a>
    <div class="d-flex gap-2">
      <a class="btn btn-outline-light btn-sm" href="index.php">Daftar</a>
      <a class="btn btn-success btn-sm" href="tambah.php">+ Tambah</a>
    </div>
  </div>
</nav>
```

9. Utilities (Class Serbaguna)

- `m-3` , `mt-2` , `mb-4` → margin (atas/bawah/samping).
- `p-3` , `py-4` , `px-2` → padding.
- `d-flex` → aktifkan flexbox.
- `justify-content-between` → bagi kiri-kanan.
- `justify-content-center` → rata tengah.
- `align-items-center` → rata vertikal.
- `gap-2` , `gap-3` → jarak antar elemen.
- `bg-light` , `bg-dark` , `bg-primary` → warna latar.

 Contoh:

```
<div class="d-flex justify-content-between align-items-center bg-light p-3">
  <span>Kiri</span>
  <span>Kanan</span>
</div>
```

✨ Kesimpulan:

Untuk proyek CRUD, class yang **paling sering dipakai** adalah:

- `btn` , `btn-*` , `btn-sm` (tombol)
- `form-control` , `mb-3` (form input)
- `table table-striped table-bordered` (tabel)
- `alert alert-*` (pesan)
- `card` , `card-header` , `card-body` (box konten)
- `navbar navbar-dark bg-dark` (navigasi atas)
- `container` , `row` , `col-md-*` (layout/grid)