

Kontrol 1 : 2 fitur utama beserta (front-end dan backend) 2 fitur prestasi siswa

1. Fitur Pencatatan Prestasi Siswa

Fitur ini memungkinkan pengguna untuk mencatat prestasi siswa dengan informasi yang terhubung ke beberapa tabel dalam basis data.

Deskripsi Fitur

Sistem Prestasi

Home

Data Siswa

Prestasi

Guru

Rekap Prestasi

Logout

Data Prestasi Siswa

✓ Tambah Data

Jumlah data: 3

No	Nama Prestasi	Juara	Tingkatan	Tahun	Aksi
1	Juara Lomba Cerdas Cermat	Juara 2	Internasional	2023	<div>Detail</div>
2	menggambar	Juara 3	Kota	2023	<div>Detail</div>
3	Lomba desain Grafis	Juara 2	Provinsi	2025	<div>Detail</div>

Pencatatan Prestasi Siswa

Nama Prestasi

Tingkatan

-- Pilih Tingkatan --

Siswa

-- Pilih Siswa --

Guru Pembimbing

-- Pilih Guru --

Tanggal Prestasi

hh/bb/tttt

Peringkat

Kategori

Umum

Tambah Prestasi

Kembali

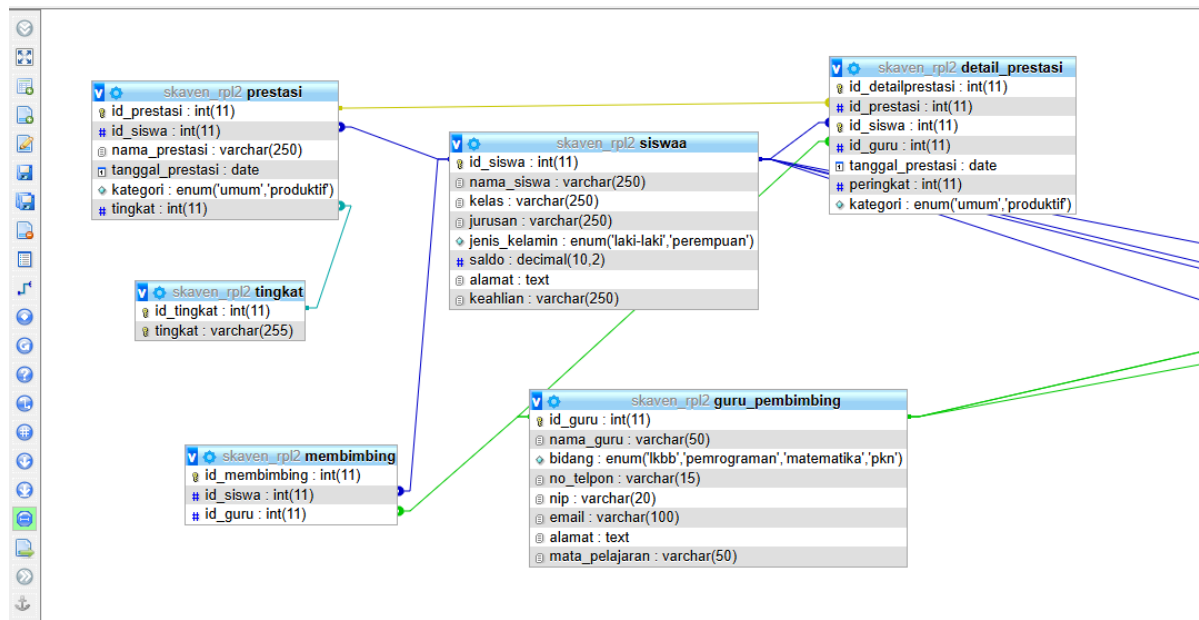
Batal

- Pencatatan Prestasi Siswa:** Memungkinkan pengguna untuk mencatat prestasi siswa, termasuk nama prestasi, tingkatan, siswa yang berprestasi, guru pembimbing, tanggal

prestasi, dan peringkat.

- **Kategori Prestasi:** Prestasi dapat dikategorikan ke dalam kategori umum atau produktif yang ditambahkan oleh pengguna.
- **Tambah Prestasi:** Fitur untuk menambahkan prestasi baru ke dalam data prestasi siswa
- **Data Prestasi Siswa:** Menampilkan daftar prestasi siswa yang telah dicatat di dalam form pencatatan prestasi siswa.

DataBase



Penjelasan database

1. Siswa (siswaa) – Data Murid yang Terdaftar di Sekolah

Setiap siswa yang ada di sekolah akan disimpan di dalam tabel **siswaa**. Tabel ini berisi informasi tentang siswa, seperti:

- Nama siswa
- Kelas dan jurusan
- Jenis kelamin
- saldo
- Alamat
- Keahlian khusus siswa (contonya: pemrograman, desain, seni, dll.)

2. Prestasi (prestasi) – Jenis Prestasi yang di raih

Setiap prestasi yang bisa diraih oleh siswa akan dicatat dalam tabel `prestasi` . Prestasi ini mencakup :

- Nama prestasi (contoh: Juara 1 Lomba Coding)
- Tanggal prestasi
- Kategori prestasi (**Umum** atau **Produktif**)
- Tingkat prestasi (**Kabupaten, Provinsi, Nasional, Internasional**)

3. Detail Prestasi (`detail_prestasi`) – Mencatat Siswa yang Meraih Prestasi

Tabel `detail_prestasi` digunakan untuk mencatat **siswa mana** yang mendapatkan prestasi tertentu. Karena satu prestasi bisa diraih oleh banyak siswa, dan satu siswa bisa mendapatkan banyak prestasi, maka kita membutuhkan tabel yang khusus untuk menyimpan informasi ini.

- **Siswa yang mendapatkan prestasi** (diambil dari tabel `siswa`)
- **Prestasi yang didapatkan** (diambil dari tabel `prestasi`)
- **Guru yang membimbing siswa tersebut**
- **Tanggal prestasi dicatat**
- **Peringkat yang diraih siswa** (misalnya: Juara 1, Juara 2, dll.)
- **Kategori prestasi** (umum/produktif)

Jadi, jika ada **5 siswa yang memenangkan lomba yang sama**, maka akan ada **5 baris data** di tabel `detail_prestasi` yang mencatat masing-masing siswa.

4. Guru Pembimbing (`guru_pembimbing`) – Guru yang Membimbing Siswa

Prestasi siswa sering kali tidak diraih sendiri, tetapi dengan bimbingan seorang guru. Oleh karena itu, setiap guru yang membimbing siswa akan dicatat dalam tabel `guru_pembimbing` .

Informasi yang disimpan di tabel ini:

- Nama guru
- Bidang keahlian (misalnya: Pemrograman, Matematika, PKN, dll.)
- Nomor telepon dan email
- Mata pelajaran yang diajarkan

5. Membimbing (`membimbing`) – Mencatat Guru yang Membimbing Siswa

Karena **satu guru bisa membimbing banyak siswa**, dan **satu siswa bisa dibimbing oleh banyak guru**, maka dibuatlah sebuah tabel **membimbing** sebagai penghubung.

- **ID Siswa** (siswa yang dibimbing)
- **ID Guru** (guru yang membimbing)

6. Tingkat (tingkat) – Prestasi yang Diraih Siswa

Setiap prestasi memiliki tingkat yang berbeda. Oleh karena itu, informasi tingkat disimpan dalam tabel **tingkat** , contohnya itu seperti ini :

- Tingkat Kabupaten
- Tingkat Provinsi
- Tingkat Nasional
- Tingkat Internasional

Rincian Fitur

Sistem Prestasi

Home

Data Siswa

Prestasi

Guru

Rekap Prestasi

Logout

Data Prestasi Siswa

✔ Tambah Data

Jumlah data: 3

No	Nama Prestasi	Juara	Tingkatan	Tahun	Aksi
1	juara Lomba Cerdas Cermat	Juara 2	Internasional	2023	<div>Detail</div>
2	menggambar	Juara 3	Kota	2023	<div>Detail</div>
3	Lomba desain Grafis	Juara 2	Provinsi	2025	<div>Detail</div>

Pencatatan Prestasi Siswa

Nama Prestasi	<input type="text"/>
Tingkatan	<input type="text" value="-- Pilih Tingkatan --"/>
Siswa	<input type="text" value="-- Pilih Siswa --"/>
Guru Pembimbing	<input type="text" value="-- Pilih Guru --"/>
Tanggal Prestasi	<input type="text" value="hh/bb/tttt"/>
Peringkat	<input type="text"/>
Kategori	<input type="text" value="Umum"/>
<div><button>Tambah Prestasi</button><button>Kembali</button><button>Batal</button></div>	

Kode Front-end

```
<!DOCTYPE html>
<html lang="id">
<head>
  <meta charset="UTF-8">
  <title>Pencatatan Prestasi Siswa</title>
  <link rel="stylesheet"
href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
>
</head>
<body>
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
  <div class="container">
    <a class="navbar-brand" href="#">Sistem Prestasi</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarNav">
      <span class="navbar-toggler-icon"></span>
    </button>
  </div>
</nav>

<div class="container mt-4">
  <h2 class="text-center"><img alt="star icon" data-bbox="392 812 415 830"/> Pencatatan Prestasi Siswa</h2>
  <form method="POST" action="">
    <div class="mb-3">
      <label>Nama Prestasi</label>
      <input type="text" name="nama_prestasi" class="form-control"
required>
    </div>
```

```

        <button type="submit" name="submit" class="btn btn-primary">Tambah
Prestasi</button>
    </form>
</div>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min
.js"></script>
</body>
</html>

```

Kode Backend

```

<?php
include "koneksi.php";

if (isset($_POST['submit'])) {
    $nama_prestasi = $_POST['nama_prestasi'];
    $tingkatan = $_POST['tingkatan'];

    // Simpan ke tabel `prestasi`
    $sql_prestasi = "INSERT INTO prestasi (nama_prestasi, tingkatan) VALUES
('$nama_prestasi', '$tingkatan')";
    if (mysqli_query($koneksi, $sql_prestasi)) {
        $id_prestasi = mysqli_insert_id($koneksi);
        $id_siswa = $_POST['id_siswa'];
        $id_guru = $_POST['id_guru'];
        $tanggal_prestasi = $_POST['tanggal_prestasi'];
        $peringkat = $_POST['peringkat'];
        $kategori = $_POST['kategori'];

        // Simpan ke tabel `detail_prestasi`
        $sql_detail = "INSERT INTO detail_prestasi (id_prestasi, id_siswa,
id_guru, tanggal_prestasi, peringkat, kategori)
VALUES ('$id_prestasi', '$id_siswa', '$id_guru',
'$tanggal_prestasi', '$peringkat', '$kategori')";

        if (mysqli_query($koneksi, $sql_detail)) {
            echo "<script>alert('Data prestasi berhasil ditambahkan!');
window.location='prestasi.php';</script>";
        } else {
            echo "Gagal menambahkan detail prestasi: " .
mysqli_error($koneksi);
        }
    } else {

```

```

        echo "Gagal menambahkan prestasi: " . mysqli_error($koneksi);
    }
}
?>

```

Penjelasan kode program

Saat pengguna mengisi dan mengirimkan formulir, data prestasi seperti **nama prestasi dan tingkatan** disimpan ke dalam tabel `prestasi`. Setelah itu, sistem mengambil **ID prestasi** yang baru ditambahkan dan menggunakan ID tersebut untuk menyimpan informasi tambahan seperti **siswa yang berprestasi, guru pembimbing, tanggal prestasi, peringkat, dan kategori** ke dalam tabel `detail_prestasi`.

Navigasi dalam program ini terdiri dari beberapa menu utama, seperti **Home, Data Siswa, Prestasi, Guru, dan Rekap Prestasi**, yang memungkinkan pengguna mengakses berbagai data terkait prestasi siswa. Formulir input memungkinkan pengguna memilih **siswa, guru pembimbing, tingkatan prestasi, serta memasukkan tanggal dan kategori prestasi**.

Setelah data disimpan, sistem menampilkan daftar prestasi dalam bentuk tabel, termasuk **nama prestasi, peringkat, tingkatan, dan tahun perolehan prestasi**. Setiap prestasi memiliki tombol "Detail" yang memungkinkan pengguna melihat informasi lebih lengkap..

Database yang digunakan disertai fitur dalam pengelolaan database

- Formulir input memungkinkan pengguna memasukkan data yang kemudian disimpan di tabel `prestasi`.
- Data tambahan seperti siswa dan guru pembimbing disimpan dalam tabel `detail_prestasi` dengan referensi ke `prestasi`.
- Tabel ini dihubungkan dengan relasi satu ke banyak, memungkinkan satu prestasi memiliki banyak detail terkait.

Relasi

Cara kerja fitur

1. Tabel `prestasi`

- Menyimpan data prestasi yang diraih oleh siswa.
- Memiliki relasi dengan `tingkat` untuk menunjukkan tingkat kejuaraan.
- Relasi dengan `detail_prestasi` untuk menghubungkan prestasi dengan siswa dan guru pembimbing.

2. Tabel `siswaa`

- Menyimpan data siswa.
- Memiliki relasi dengan `detail_prestasi` untuk mencatat siswa yang meraih prestasi.
- Relasi dengan `membimbing` untuk menunjukkan siswa yang dibimbing oleh guru.

3. Tabel `guru_pembimbing`

- Menyimpan data guru yang membimbing siswa dalam meraih prestasi.
- Memiliki relasi dengan `detail_prestasi` untuk mencatat guru yang membimbing dalam prestasi tertentu.
- Relasi dengan `membimbing` untuk mencatat hubungan antara guru dan siswa.

4. Tabel `detail_prestasi`

- Berperan sebagai tabel penghubung antara `prestasi`, `siswaa`, dan `guru_pembimbing`.
- Menyimpan informasi detail tentang prestasi yang diraih, seperti tanggal, peringkat, dan kategori.

5. Tabel `tingkat`

- Berisi informasi tentang tingkat kejuaraan (misalnya kabupaten, provinsi, nasional).
- Memiliki relasi dengan `prestasi` untuk menunjukkan tingkatan prestasi yang dicapai.

6. Tabel `membimbing`

- Berfungsi sebagai tabel penghubung antara `siswaa` dan `guru_pembimbing`.
- Mencatat guru yang membimbing siswa tertentu.

```
$query = "SELECT dp.id_detailprestasi, p.nama_prestasi, dp.peringkat AS
juara, p.tingkat, YEAR(dp.tanggal_prestasi) AS tahun

FROM detail_prestasi dp

JOIN prestasi p ON dp.id_prestasi = p.id_prestasi";
```

Tujuan: Query ini bertujuan untuk mengambil informasi rinci tentang prestasi, termasuk ID detail, nama prestasi, peringkat, tingkat, dan tahun dari tanggal prestasi.

1. **SELECT Clause:**

- `dp.id_detailprestasi`: Mengambil ID dari tabel `detail_prestasi`, yang mungkin merupakan kunci utama untuk setiap entry prestasi.

- `p.nama_prestasi` : Mengambil nama prestasi dari tabel `prestasi` . - `dp.peringkat AS juara` : Mengambil peringkat dari tabel `detail_prestasi` dan memberinya alias `juara` , yang menunjukkan posisi atau ranking. - `p.tingkat` : Mengambil tingkat dari tabel `prestasi` , yang bisa menunjukkan level atau kategori prestasi (misalnya: SD, SMP, SMA).
- `YEAR(dp.tanggal_prestasi) AS tahun` : Mengambil tahun dari kolom `tanggal_prestasi` , memberikan alias `tahun` . Ini berguna untuk analisis berdasarkan tahun.

2. FROM Clause:

- `FROM detail_prestasi dp` : Menentukan tabel utama yang digunakan dalam query, yaitu `detail_prestasi` , dengan alias `dp` untuk menyederhanakan penulisan.

3. JOIN Clause:

- `JOIN prestasi p ON dp.id_prestasi = p.id_prestasi` : Melakukan join antara tabel `detail_prestasi` dan tabel `prestasi` berdasarkan kolom `id_prestasi` . Ini memungkinkan pengambilan data dari kedua tabel, menghubungkan detail prestasi dengan informasi umum tentang prestasi tersebut.

```
$query = "SELECT dp.*, p.nama_prestasi, p.tingkat, s.nama_siswa AS
nama_siswa, g.nama_guru

FROM detail_prestasi dp

JOIN prestasi p ON dp.id_prestasi = p.id_prestasi

JOIN siswaa s ON dp.id_siswa = s.id_siswa

JOIN guru_pembimbing g ON dp.id_guru = g.id_guru

WHERE dp.id_detailprestasi = '$id_detail';"
```

Tujuan: Query ini bertujuan untuk mengambil informasi lengkap tentang detail prestasi tertentu, termasuk nama prestasi, tingkat, nama siswa, dan nama guru pembimbing.

1. SELECT Clause:

- `dp.*` : Mengambil semua kolom dari tabel `detail_prestasi` . Ini akan memberikan semua informasi terkait detail prestasi.
- `p.nama_prestasi` : Mengambil nama prestasi dari tabel `prestasi` .
- `p.tingkat` : Mengambil tingkat prestasi dari tabel `prestasi` .
- `s.nama_siswa AS nama_siswa` : Mengambil nama siswa dari tabel `siswaa` dan

memberinya alias `nama_siswa`.

- `g.nama_guru` : Mengambil nama guru dari tabel `guru_pembimbing`.

2. FROM Clause:

- `FROM detail_prestasi dp` : Menentukan tabel utama yang digunakan dalam query, yaitu `detail_prestasi`, dengan alias `dp`.

3. JOIN Clauses:

- `JOIN prestasi p ON dp.id_prestasi = p.id_prestasi` : Menghubungkan tabel `detail_prestasi` dengan tabel `prestasi` berdasarkan kolom `id_prestasi`.
- `JOIN siswaa s ON dp.id_siswa = s.id_siswa` : Menghubungkan tabel `detail_prestasi` dengan tabel `siswaa` untuk mendapatkan informasi siswa berdasarkan `id_siswa`.
- `JOIN guru_pembimbing g ON dp.id_guru = g.id_guru` : Menghubungkan tabel `detail_prestasi` dengan tabel `guru_pembimbing` untuk mendapatkan informasi guru berdasarkan `id_guru`.

4. WHERE Clause:

- `WHERE dp.id_detailprestasi = '$id_detail'` : Menentukan kondisi untuk filter hasil, hanya mengambil data yang memiliki `id_detailprestasi` sesuai dengan nilai `$id_detail`. Ini biasanya digunakan untuk menampilkan detail dari satu prestasi spesifik.

Agregasi (tidak ada)

Autentikasi

- **Admin** → Memiliki akses penuh untuk mengelola data pengguna, siswa, guru, dan prestasi.
- **Guru** → Hanya bisa mengakses data siswa dan mencatat prestasi.

Analisis Program Autentikasi

```
<?php
session_start();
include "koneksi.php";
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $username = $_POST['username'];
    $password = md5($_POST['password']); // Hash password agar cocok dengan
    database
    $query = "SELECT * FROM users WHERE username='$username' AND
    password='$password'";
    $result = mysqli_query($koneksi, $query);
```

```

        if (mysqli_num_rows($result) == 1) {
            $_SESSION['username'] = $username;
            header("Location: index.php"); // Redirect ke halaman utama
        } else {
            echo "<script>alert('Username atau password salah!');
window.location='login.php';</script>";
        }
    }
    ?>

<?php
session_start();
include "koneksi.php";
if (!isset($_SESSION['username'])) {
    header("Location: login.php");
    exit();
}
?>

```

1. Penerimaan Data dari Pengguna

Ketika pengguna mengisi formulir login dengan **username** dan **password**, sistem akan menerima data tersebut untuk diperiksa di database.

2. Enkripsi Password

Sebelum mencocokkan password yang dimasukkan dengan yang ada di database, sistem mengenkripsi password.

3. Pengecekan Data di Database

Setelah menerima username dan password, sistem akan mencari data tersebut di dalam database. Jika ada **kecocokan**, maka pengguna akan diberikan akses ke dalam sistem.

4. Pengalihan ke Halaman Utama

Setelah login berhasil, pengguna akan diarahkan ke halaman utama sistem agar bisa mulai menggunakan fitur yang tersedia

2. Fitur Manajemen Data Guru

Fitur manajemen data guru adalah bagian dari sistem informasi sekolah yang memungkinkan pengguna untuk mengelola informasi guru secara efisien. Fitur ini mencakup penambahan, pengeditan, dan penghapusan data guru, serta pencatatan jumlah guru berdasarkan bidang ajar.

Deskripsi Fitur

Sistem Guru Home Data Siswa Prestasi Guru


Data Guru

✓ Tambah Data

📌 Jumlah Guru Per Bidang (Bidang dengan lebih dari 1 guru)

Bidang	Jumlah Guru
pkn	2

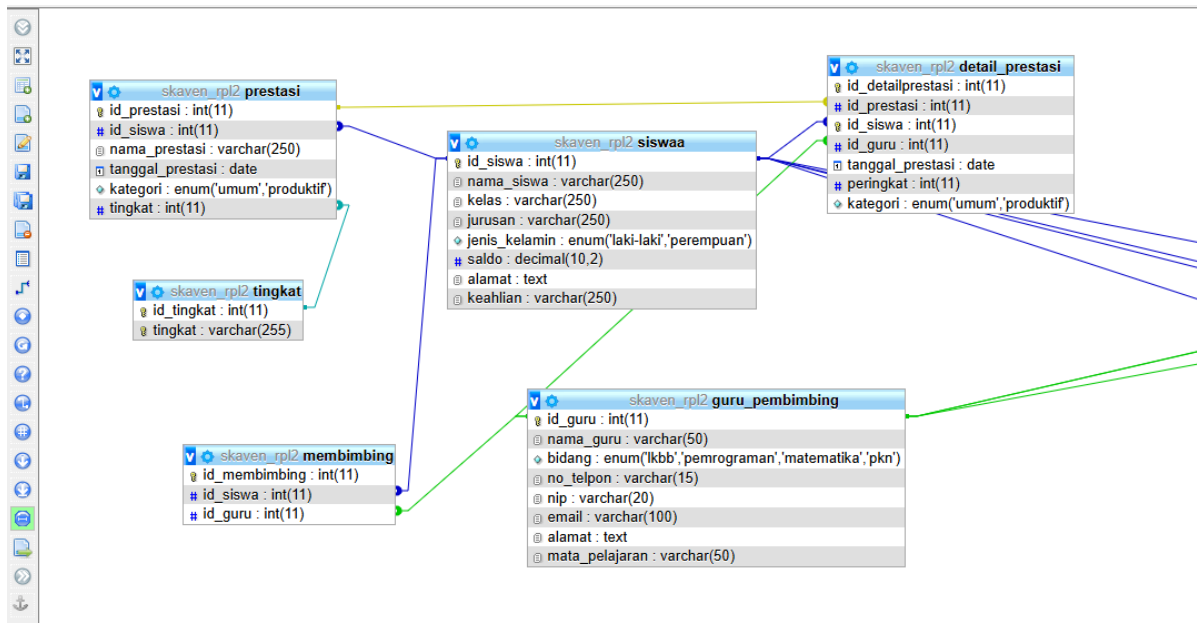
Data Lengkap Guru

ID	Nama	Bidang	No HP	Email	Alamat	Mata Pelajaran	Tindakan
1	ibu anty	pemrograman	08765487	antyrpl@gmail.com	jl.ince nurdin	pemrong	 Edit  Hapus
2	ibu tenri	matematika	0879654321	tenritte23@gmail.com	jl.rajawali	matematika	 Edit  Hapus
3	pak ibe	pkn	08765478	ibrahim12@gmail.com	jl.cendrawasih	basis data	 Edit  Hapus
4	ibu asrah	pkn	098765876	asrahwati@gmail.com	jl.borong	pkk	 Edit  Hapus
5	pak fajar	lkbb	0876544598	fajarji@gmail.com	jl.sudirman	web	 Edit  Hapus

Seperti yang ada pada gambar diatas ini menunjukkan front-end dari fitur manajemen data guru. Berikut adalah beberapa elemen UI yang digunakan:

1. **Menambahkan Data Guru** - Pengguna dapat memasukkan informasi guru baru seperti nama, bidang, nomor HP, email, alamat, dan mata pelajaran yang diajarkan.
2. **Menampilkan Data Guru** - Sistem menampilkan daftar guru dalam bentuk tabel yang mencakup semua informasi penting.
3. **Mengedit Data Guru** - Pengguna dapat memperbarui data guru yang sudah ada melalui tombol "Edit".
4. **Menghapus Data Guru** - Pengguna dapat menghapus data guru yang tidak diperlukan dengan tombol "Hapus".
5. **Menampilkan Jumlah Guru per Bidang** - Sistem menampilkan jumlah guru berdasarkan bidang yang memiliki lebih dari satu pengajar.
6. **Hubungan antar tabel** - Data guru terhubung dengan tabel lain seperti tabel pengguna dan tabel mata pelajaran.
7. **Form Tambah Data**: Tombol "Tambah Data" berwarna biru untuk menambahkan data guru baru.
8. **Statistik Jumlah Guru**: Menampilkan jumlah guru per bidang dalam format tabel.
9. **Tabel Data Guru**: Berisi daftar guru dengan kolom ID, Nama, Bidang, No HP, Email, Alamat, Mata Pelajaran, serta tombol "Edit" dan "Hapus".
10. **Tombol Aksi**:
 - **Tombol Edit** berwarna kuning
 - **Tombol Hapus** berwarna merah

Database



Tabel `guru_pembimbing` digunakan dalam fitur manajemen data guru untuk menyimpan informasi tentang guru yang membimbing siswa dalam prestasi umum atau produktif.

1. Menyimpan Data Guru

- Data setiap guru dicatat dengan atribut seperti **nama**, **bidang**, **nomor telepon**, dan **mata pelajaran**.

1. Menampilkan Daftar Guru

- Sistem dapat menampilkan semua guru beserta informasi mereka dalam tabel.

3. Menambahkan Guru Baru

- Memungkinkan admin untuk menambahkan guru baru dalam sistem.

4. Mengedit dan Menghapus Data Guru

- Jika ada perubahan, data guru dapat diperbarui atau dihapus.

5. Menghubungkan Guru dengan Siswa

- Dalam fitur ini, guru bisa dihubungkan dengan siswa melalui tabel `membimbing` untuk mencatat siapa yang membimbing siapa.

Rincian Fitur

Sistem Guru Home Data Siswa Prestasi Guru

Data Guru

✓ Tambah Data

📌 Jumlah Guru Per Bidang (Bidang dengan lebih dari 1 guru)

Bidang	Jumlah Guru
pkn	2

Data Lengkap Guru

ID	Nama	Bidang	No HP	Email	Alamat	Mata Pelajaran	Tindakan
1	ibu anty	pemrograman	08765487	antyrpl@gmail.com	jl.ince nurdin	pemrong	 Edit  Hapus
2	ibu tenri	matematika	0879654321	tenritte23@gmail.com	jl.rajawali	matematika	 Edit  Hapus
3	pak ibe	pkn	08765478	ibrahim12@gmail.com	jl.cendrawasih	basis data	 Edit  Hapus
4	ibu asrah	pkn	098765876	asrahwati@gmail.com	jl.borong	pkk	 Edit  Hapus
5	pak fajar	lkbb	0876544598	fajarji@gmail.com	jl.sudirman	web	 Edit  Hapus

Backend

```
<?php
include 'koneksi.php';

// Query untuk menghitung jumlah guru per bidang (hanya bidang dengan lebih
dari 1 guru)
$query_group = "SELECT bidang, COUNT(*) AS jumlah_guru
                FROM guru_pembimbing
                GROUP BY bidang
                HAVING COUNT(*) > 1";
$result_group = mysqli_query($koneksi, $query_group);

// Query untuk mengambil semua data guru
$query_all = "SELECT * FROM guru_pembimbing";
$result_all = mysqli_query($koneksi, $query_all);
?>
```

Front-end

```
<!DOCTYPE html>
<html lang="id">
<head>
    <title>Data Guru</title>
    <link rel="stylesheet"
```

```

href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
>
</head>
<body>

<!-- Navbar -->
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
  <div class="container">
    <a class="navbar-brand" href="#">Sistem Guru</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarNav">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav">
        <li class="nav-item">
          <a class="nav-link" href="index.php">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="tbl_siswa.php">Data Siswa</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="prestasi.php">Prestasi</a>
        </li>
        <li class="nav-item">
          <a class="nav-link active" href="tbl_guru.php">Guru</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="rekap_prestasi.php">Rekap
Prestasi</a>
        </li>
      </ul>
    </div>
  </div>
</nav>

<!-- Konten -->
<div class="container mt-4">
  <h2 class="text-center">Data Guru</h2>
  <a href="tambah_guru.php" class="btn btn-primary mb-3">✅ Tambah Data</a>

  <!-- Menampilkan jumlah guru per bidang -->
  <h3>🔥 Jumlah Guru Per Bidang (Bidang dengan lebih dari 1 guru)</h3>
  <table class="table table-bordered">
    <tr>
      <th>Bidang</th>

```

```

        <th>Jumlah Guru</th>
    </tr>
    <?php while ($data = mysqli_fetch_assoc($result_group)) { ?>
    <tr>
        <td><?= $data['bidang']; ?></td>
        <td><?= $data['jumlah_guru']; ?></td>
    </tr>
    <?php } ?>
</table>

<!-- Menampilkan Data Lengkap Guru -->
<h3>📋 Data Lengkap Guru</h3>
<table class="table table-bordered">
    <tr>
        <th>ID</th>
        <th>Nama</th>
        <th>Bidang</th>
        <th>No HP</th>
        <th>Email</th>
        <th>Alamat</th>
        <th>Mata Pelajaran</th>
        <th>Tindakan</th>
    </tr>
    <?php while ($data = mysqli_fetch_assoc($result_all)) { ?>
    <tr>
        <td><?php echo $data['id_guru']; ?></td>
        <td><?php echo $data['nama_guru']; ?></td>
        <td><?php echo $data['bidang']; ?></td>
        <td><?php echo $data['no_telpon']; ?></td>
        <td><?php echo $data['email']; ?></td>
        <td><?php echo $data['alamat']; ?></td>
        <td><?php echo $data['mata_pelajaran']; ?></td>
        <td>
            <a href="edit_guru.php?id=<?= $data['id_guru']; ?>" class="btn
btn-warning btn-sm"> ✎ Edit</a>
            <a href="hapus_guru.php?id=<?= $data['id_guru']; ?>"
class="btn btn-danger btn-sm"
onclick="return confirm('Apakah anda yakin ingin menghapus
data ini?')"> 🗑 Hapus</a>
        </td>
    </tr>
    <?php } ?>
</table>
</div>

<script

```



```
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min
.js"></script>
</body>
</html>
```

Penjelasan kode program

Data guru diambil dari tabel `guru_pembimbing`, kemudian ditampilkan dalam bentuk tabel interaktif. Sistem juga melakukan pengelompokan data berdasarkan bidang untuk mengetahui jumlah guru di setiap bidang yang memiliki lebih dari satu guru. Selain itu, fitur **CRUD (Create, Read, Update, Delete)** memungkinkan pengguna menambah, mengedit, dan menghapus data guru. Navbar memudahkan navigasi antara halaman, sementara tabel yang ditampilkan berisi informasi guru seperti **ID, Nama, Bidang, No HP, Email, Alamat, dan Mata Pelajaran**. Tombol **Edit** dan **Hapus** memberikan kemudahan dalam pengelolaan data, dengan konfirmasi sebelum penghapusan dilakukan untuk mencegah kesalahan.

Database yang digunakan disertai fitur dalam pengelolaan database

Tabel `guru_pembimbing`

- **id_guru** (*INT, PRIMARY KEY, AUTO_INCREMENT*) → Sebagai identitas unik untuk setiap guru.
- **nama_guru** (*VARCHAR*) → Menyimpan nama guru.
- **bidang** (*VARCHAR*) → Menyimpan bidang keahlian guru.
- **no_telpon** (*VARCHAR*) → Nomor telepon guru untuk komunikasi.
- **email** (*VARCHAR*) → Email guru untuk keperluan informasi.
- **alamat** (*TEXT*) → Alamat tempat tinggal guru.
- **mata_pelajaran** (*VARCHAR*) → Mata pelajaran yang diajarkan oleh guru.

1. Menampilkan Data Guru

- Sistem mengambil seluruh data dari tabel `guru_pembimbing` dan menampilkannya dalam tabel interaktif di halaman web.
- Pengguna dapat melihat informasi guru secara lengkap dan terstruktur.

2. Mengelompokkan Guru Berdasarkan Bidang

- Sistem menghitung jumlah guru berdasarkan bidangnya.
- Jika suatu bidang memiliki lebih dari satu guru, maka bidang tersebut ditampilkan pada bagian "Jumlah Guru Per Bidang".
- Fitur ini membantu dalam perencanaan tenaga pengajar agar distribusi guru lebih merata.

3. Menambah Data Guru

- Pengguna dapat menambahkan guru baru melalui formulir input.
- Data yang dimasukkan akan disimpan ke dalam tabel `guru_pembimbing`.

4. Mengedit Data Guru

- Fitur ini memungkinkan pengguna memperbarui informasi guru yang sudah ada, seperti mengganti nomor HP, email, atau mata pelajaran yang diajarkan.
- Sistem mengidentifikasi data berdasarkan **id_guru** untuk memastikan perubahan yang dilakukan sesuai dengan guru yang bersangkutan.

5. Menghapus Data Guru

- Pengguna dapat menghapus data guru yang sudah tidak diperlukan.
- Sistem meminta konfirmasi sebelum menghapus untuk menghindari kesalahan penghapusan data penting.

Relasi (tidak ada)

Agregasi

Tujuan

Agregasi memungkinkan pengelompokkan data guru berdasarkan bidang keahlian. Dengan menghitung jumlah guru per bidang, manajemen dapat dengan mudah memahami distribusi tenaga pengajar di berbagai bidang mata pelajaran. Ini membantu dalam analisis kebutuhan pengajaran di setiap bidang.

```
$query_group = "SELECT bidang, COUNT(*) AS jumlah_guru
                FROM guru_pembimbing
                GROUP BY bidang
                HAVING COUNT(*) > 1";
```

1. **SELECT bidang, COUNT(*) AS jumlah_guru:**

- Mengambil kolom `bidang` dan menghitung jumlah guru di setiap bidang. `COUNT(*)` menghitung jumlah baris dalam setiap kelompok.

2. **FROM guru_pembimbing:**

- Menentukan tabel yang digunakan untuk mengambil data, yaitu `guru_pembimbing`.

3. GROUP BY bidang:

- Mengelompokkan hasil berdasarkan kolom `bidang`. Setiap bidang yang berbeda akan menjadi satu kelompok.

4. HAVING COUNT(*) > 1:

- Menyaring hasil untuk hanya menampilkan bidang yang memiliki lebih dari satu guru. Ini adalah langkah penting karena memungkinkan kita untuk fokus pada bidang yang memiliki lebih banyak tenaga pengajar.

Autentikasi

- **Guru** → Hanya bisa mengakses data siswa dan mencatat prestasi.

Analisis Program Autentikasi

```
<?php
session_start();
include "koneksi.php";
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $username = $_POST['username'];
    $password = md5($_POST['password']); // Hash password agar cocok dengan
    database
    $query = "SELECT * FROM users WHERE username='$username' AND
    password='$password'";
    $result = mysqli_query($koneksi, $query);
    if (mysqli_num_rows($result) == 1) {
        $_SESSION['username'] = $username;
        header("Location: index.php"); // Redirect ke halaman utama
    } else {
        echo "<script>alert('Username atau password salah!');
        window.location='login.php';</script>";
    }
}
?>
```

```
<?php
session_start();
include "koneksi.php";
if (!isset($_SESSION['username'])) {
    header("Location: login.php");
    exit();
}
```

```
}  
?>
```

1. Penerimaan Data dari Pengguna

Ketika pengguna mengisi formulir login dengan **username** dan **password**, sistem akan menerima data tersebut untuk diperiksa di database.

2. Enkripsi Password

Sebelum mencocokkan password yang dimasukkan dengan yang ada di database, sistem mengenkripsi password.

3. Pengecekan Data di Database

Setelah menerima username dan password, sistem akan mencari data tersebut di dalam database. Jika ada **kecocokan**, maka pengguna akan diberikan akses ke dalam sistem.

4. Pengalihan ke Halaman Utama

Setelah login berhasil, pengguna akan diarahkan ke halaman utama sistem agar bisa mulai menggunakan fitur yang tersedia

Kontrol 2 : 3 fitur utama beserta (front-end dan backend) 2 fitur prestasi siswa

3. Fitur Ringkasan Prestasi siswa

Fitur "Ringkasan Prestasi" bertujuan untuk memberikan informasi terperinci mengenai prestasi siswa dalam format yang mudah dipahami. Fitur ini menampilkan statistik jumlah prestasi per tingkatan, siswa dengan prestasi terbanyak, dan daftar prestasi terbaru yang diraih oleh siswa.

Deskripsi Fitur

Ringkasan Prestasi

Jumlah Prestasi per Tingkatan

Tingkatan	Jumlah Prestasi
Internasional	1
Kota	1
Provinsi	1

Siswa dengan prestasi terbaru

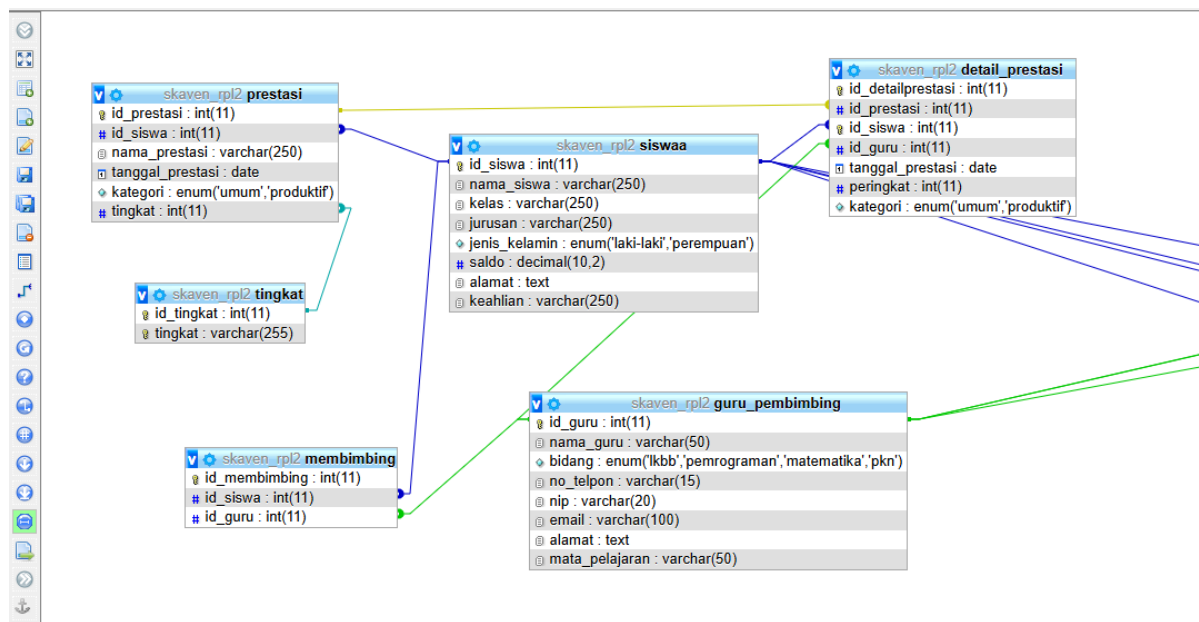
Nama Siswa	Jumlah Prestasi
Suci Azizah	1
condrado	1

Daftar Prestasi Terbaru

Nama Siswa	Nama Prestasi	Tingkatan	Tanggal Perolehan
Qalby	Lomba desain Grafis	Provinsi	20-03-2025
condrado	juara Lomba Cerdas Cermat	Internasional	04-03-2023
Suci Azizah	menggambar	Kota	12-02-2023

- **Data Siswa:** Menyimpan dan mengelola informasi siswa, termasuk nama, kelas, dan prestasi yang diperoleh.
- **Data Guru:** Mengelola informasi tentang guru, seperti nama, bidang keahlian, dan kontak.
- **Prestasi:** Mencatat prestasi yang diraih oleh siswa dan mengaitkannya dengan guru pembimbing.
- **Rekap Prestasi:** Menampilkan ringkasan statistik prestasi, termasuk jumlah prestasi per tingkatan dan siswa dengan prestasi terbanyak.
- **Autentikasi Pengguna:** Mengelola akses pengguna berdasarkan jenis peran (admin, guru)

Database



- **siswaa**: Tabel ini menyimpan informasi tentang siswa, termasuk ID, nama, kelas, dan informasi kontak.
- **guru_pembimbing**: Menyimpan data guru, termasuk ID, nama, bidang keahlian, dan kontak.
- **prestasi**: Menyimpan informasi tentang prestasi, termasuk ID, nama prestasi, tingkat, dan kategori.
- **detail_prestasi**: Menyimpan detail tentang perolehan prestasi oleh siswa, mengaitkan siswa, guru, dan prestasi.
- **membimbing**: Menyimpan hubungan antara guru dan prestasi yang mereka bimbing.

Rincian Fitur

Sistem Guru

Home


Data Siswa

Prestasi


Guru

Rekap Prestasi

Logout




Ringkasan Prestasi




Jumlah Prestasi per Tingkatan

Tingkatan	Jumlah Prestasi
Internasional	1
Kota	1
Provinsi	1







Siswa dengan prestasi terbaru

Nama Siswa	Jumlah Prestasi
Suci Azizah	1
condrado	1



Daftar Prestasi Terbaru

 Nama Siswa	 Nama Prestasi	 Tingkatan	 Tanggal Perolehan
Qalby	Lomba desain Grafis	Provinsi	20-03-2025
condrado	juara Lomba Cerdas Cermat	Internasional	04-03-2023
Suci Azizah	menggambar	Kota	12-02-2023

Kode front-end

```
<!DOCTYPE html>
<html lang="id">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>  Rekap Prestasi</title>

  <!-- Bootstrap CSS -->
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body class="bg-light">
```

```

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
  <div class="container">
    <a class="navbar-brand" href="#">Sistem Guru</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarNav">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav">
        <li class="nav-item"><a class="nav-link"
href="index.php">Home</a></li>
        <li class="nav-item"><a class="nav-link"
href="tbl_siswa.php">Data Siswa</a></li>
        <li class="nav-item"><a class="nav-link"
href="prestasi.php">Prestasi</a></li>
        <li class="nav-item"><a class="nav-link"
href="tbl_guru.php">Guru</a></li>
        <li class="nav-item"><a class="nav-link active"
href="rigkas_prestasi.php">Rekap Prestasi</a></li>
      </ul>
    </div>
  </div>
</nav>

<div class="container mt-5">
  <h2 class="text-center mb-4"><img alt="Ringkasan Prestasi icon" data-bbox="438 535 458 550"/> Ringkasan Prestasi</h2>

  <div class="row mb-4">
    <!-- Jumlah Prestasi per Tingkatan -->
    <div class="col-md-6">
      <div class="card shadow-sm">
        <div class="card-header bg-primary text-white">
          <h5 class="mb-0"><img alt="Ringkasan Prestasi icon" data-bbox="478 675 498 690"/> Jumlah Prestasi per Tingkatan</h5>
        </div>
        <div class="card-body">
          <table class="table table-striped table-bordered">
            <thead class="table-blue">
              <tr>
                <th>Tingkatan</th>
                <th>Jumlah Prestasi</th>
              </tr>
            </thead>
            <tbody>
              <?php while ($row =
mysqli_fetch_assoc($result_tingkat)) : ?>
                <tr>

```

```

                <td><?= $row['tingkat'] ?></td>
                <td><?= $row['jumlah'] ?></td>
            </tr>
        <?php endwhile; ?>
    </tbody>
</table>
</div>
</div>
</div>

<!-- Siswa dengan Prestasi Terbanyak -->
<div class="col-md-6">
    <div class="card shadow-sm">
        <div class="card-header bg-success text-white">
            <h5 class="mb-0">🏆 Siswa dengan Prestasi Terbanyak</h5>
        </div>
        <div class="card-body text-center">
            <h4 class="text-primary"><?= $top_siswa['nama_siswa'] ??
"Belum Ada" ?></h4>
            <p class="mb-0"><strong><?= $top_siswa['jumlah'] ?? 0 ?>
</strong> Prestasi</p>
        </div>
    </div>
</div>
</div>

<!-- Daftar Prestasi Terbaru -->
<div class="card shadow-sm">
    <div class="card-header bg-info text-white">
        <h5 class="mb-0">📅 Daftar Prestasi Terbaru</h5>
    </div>
    <div class="card-body">
        <table class="table table-striped table-bordered">
            <thead class="table-blue">
                <tr>
                    <th>👤 Nama Siswa</th>
                    <th>🏆 Nama Prestasi</th>
                    <th>📊 Tingkatan</th>
                    <th>📅 Tanggal Perolehan</th>
                    <th>👨‍🏫 Guru Pembimbing</th>
                </tr>
            </thead>
            <tbody>
                <?php while ($row = mysqli_fetch_assoc($result_recent)) :
?>
                    <tr>

```



```

        <td><?= $row['nama_siswa'] ?></td>
        <td><?= $row['nama_prestasi'] ?></td>
        <td><?= $row['tingkat'] ?></td>
        <td><?= date('d-m-Y',
strtotime($row['tanggal_prestasi'])) ?></td>
        <td><?= $row['nama_guru'] ?? 'Tidak Ada' ?></td>
    </tr>
    <?php endwhile; ?>
</tbody>
</table>
</div>
</div>

<!-- Bootstrap JS -->
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min
.js"></script>
</body>
</html>

```

Kode Backend

```

<?php
include "koneksi.php"; // Pastikan koneksi ke database

// Ambil jumlah prestasi berdasarkan tingkat
$query_tingkat = "SELECT p.tingkat, COUNT(dp.id_prestasi) AS jumlah
                  FROM prestasi p
                  JOIN detail_prestasi dp ON p.id_prestasi = dp.id_prestasi
                  GROUP BY p.tingkat";
$result_tingkat = mysqli_query($koneksi, $query_tingkat);

// Ambil siswa dengan prestasi terbanyak
$query_top_siswa = "SELECT s.nama_siswa, COUNT(dp.id_prestasi) AS jumlah
                   FROM siswaa s
                   JOIN detail_prestasi dp ON s.id_siswa = dp.id_siswa
                   GROUP BY s.nama_siswa
                   ORDER BY jumlah DESC
                   LIMIT 1";
$result_top_siswa = mysqli_query($koneksi, $query_top_siswa);
$top_siswa = mysqli_fetch_assoc($result_top_siswa);

// Ambil daftar prestasi terbaru dengan guru pembimbing

```

```
$query_recent = "SELECT s.nama_siswa AS nama_siswa, p.nama_prestasi,
p.tingkat, dp.tanggal_prestasi, g.nama_guru
FROM detail_prestasi dp
JOIN siswaa s ON dp.id_siswa = s.id_siswa
JOIN prestasi p ON dp.id_prestasi = p.id_prestasi
LEFT JOIN membimbing m ON dp.id_prestasi = m.id_prestasi
LEFT JOIN guru_pembimbing g ON m.id_guru = g.id_guru
ORDER BY dp.tanggal_prestasi DESC
LIMIT 10";

$result_recent = mysqli_query($koneksi, $query_recent);
?>
```

Analisis Penjelasan Kode Program

- **Ambil Jumlah Prestasi per Tingkatan:** Query ini menghitung jumlah prestasi yang diraih untuk setiap tingkatan dan mengelompokkannya berdasarkan tingkatan.
- **Ambil Siswa dengan Prestasi Terbanyak:** Query ini mengambil nama siswa dan jumlah prestasi yang diraih oleh siswa tersebut, mengurutkannya berdasarkan jumlah prestasi dan membatasi hasil pada satu siswa dengan prestasi terbanyak.
- **Ambil Daftar Prestasi Terbaru:** Query ini mengambil informasi terbaru mengenai prestasi siswa, termasuk nama siswa, nama prestasi, tingkatan, tanggal perolehan, dan guru pembimbing.

Database yang Digunakan:

- **prestasi:** Menyimpan informasi tentang prestasi.
- **detail_prestasi:** Menyimpan detail perolehan prestasi yang mengaitkan siswa dan prestasi.
- **siswaa:** Menyimpan informasi tentang siswa.
- **guru_pembimbing:** Menyimpan informasi tentang guru yang membimbing siswa dalam prestasi.

Peran Fitur dalam Mengelola Database:

- Fitur ini berfungsi untuk menyajikan informasi ringkas dan terstruktur mengenai prestasi siswa. Ini membantu manajemen dalam menganalisis performa siswa, mengidentifikasi siswa berprestasi, dan memantau perkembangan prestasi di berbagai tingkatan. Dengan informasi yang disajikan, pihak manajemen dapat mengambil keputusan yang lebih tepat terkait dukungan dan pengembangan program pengajaran.

Relasi

Tujuan Mengaitkan siswa dengan prestasi yang mereka capai dan guru yang membimbing mereka, sehingga informasi dapat diakses secara menyeluruh.

```
$query_tingkat = "SELECT p.tingkat, COUNT(dp.id_prestasi) AS jumlah
                  FROM prestasi p
                  JOIN detail_prestasi dp ON p.id_prestasi = dp.id_prestasi
                  GROUP BY p.tingkat";
```

- **Relasi Tabel:** Tabel `prestasi` dan `detail_prestasi` dihubungkan melalui `id_prestasi`.
- **Analisis:** Query ini menghitung jumlah prestasi per tingkatan. Dengan menggunakan `JOIN`, kita mendapatkan informasi dari kedua tabel, dan `GROUP BY` mengelompokkan hasil berdasarkan tingkatan, sehingga kita bisa melihat berapa banyak prestasi yang diraih di setiap tingkatan.

```
$query_top_siswa = "SELECT s.nama_siswa, COUNT(dp.id_prestasi) AS jumlah
                   FROM siswaa s
                   JOIN detail_prestasi dp ON s.id_siswa = dp.id_siswa
                   GROUP BY s.nama_siswa
                   ORDER BY jumlah DESC
                   LIMIT 1";
```

- **Relasi Tabel:** Tabel `siswaa` dan `detail_prestasi` dihubungkan melalui `id_siswa`.
- **Analisis:** Query ini menghitung jumlah prestasi yang diraih oleh masing-masing siswa. Dengan `JOIN`, informasi siswa dan prestasi mereka digabungkan, dan `ORDER BY` digunakan untuk mengurutkan siswa berdasarkan jumlah prestasi yang diraih, menampilkan siswa dengan prestasi terbanyak di bagian atas.

```
$query_recent = "SELECT s.nama_siswa AS nama_siswa, p.nama_prestasi,
p.tingkat, dp.tanggal_prestasi, g.nama_guru
                FROM detail_prestasi dp
                JOIN siswaa s ON dp.id_siswa = s.id_siswa
                JOIN prestasi p ON dp.id_prestasi = p.id_prestasi
                LEFT JOIN membimbing m ON dp.id_prestasi = m.id_prestasi
                LEFT JOIN guru_pembimbing g ON m.id_guru = g.id_guru
                ORDER BY dp.tanggal_prestasi DESC
                LIMIT 10";
```

- **Relasi Tabel:** Tabel `detail_prestasi`, `siswaa`, `prestasi`, `membimbing`, dan `guru_pembimbing` dihubungkan.
- **Analisis:** Query ini mengumpulkan informasi tentang prestasi terbaru yang diraih oleh siswa, termasuk nama siswa, nama prestasi, tingkatan, tanggal perolehan, dan nama guru pembimbing. Pada bagian ini, `LEFT JOIN` digunakan untuk memastikan bahwa semua prestasi ditampilkan meskipun tidak semua prestasi dihubungkan dengan guru pembimbing.

Cara Kerja Fitur terhadap Relasi

1. Ketika fitur diakses, semua query dijalankan untuk mengambil data yang diperlukan dari database.
2. **Pengelompokan dan Penghitungan:** Data dari tabel-tabel yang terhubung diproses untuk menghitung jumlah prestasi per tingkatan dan mengidentifikasi siswa dengan prestasi terbanyak.
3. **Penyajian Data:** Data yang diperoleh kemudian disajikan dalam format tabel yang mudah dipahami di antarmuka pengguna. Ini memberikan gambaran yang jelas tentang prestasi siswa dan membantu dalam pengambilan keputusan manajerial.

Agregasi

Tujuan

Menghitung jumlah prestasi berdasarkan tingkatan untuk memberikan gambaran yang jelas tentang distribusi prestasi di berbagai tingkatan. Ini membantu manajemen dalam memahami area di mana siswa mungkin unggul atau perlu lebih banyak dukungan.

1. Jumlah Prestasi Berdasarkan Tingkat:

```
$query_tingkat = "SELECT p.tingkat, COUNT(dp.id_prestasi) AS jumlah
                  FROM prestasi p
                  JOIN detail_prestasi dp ON p.id_prestasi = dp.id_prestasi
                  GROUP BY p.tingkat";
```

- **Fungsi:** COUNT(dp.id_prestasi) menghitung jumlah prestasi untuk setiap tingkatan.
- **Relasi:** Tabel prestasi dan detail_prestasi dihubungkan melalui id_prestasi. Ini memastikan bahwa data prestasi yang dihitung terkait dengan tingkatan yang benar.
- **Pengelompokan:** GROUP BY p.tingkat mengelompokkan hasil berdasarkan tingkatan, sehingga setiap tingkatan ditampilkan dengan jumlah prestasi yang sesuai.

2. Siswa dengan Prestasi Terbanyak

```
$query_top_siswa = "SELECT s.nama_siswa, COUNT(dp.id_prestasi) AS jumlah
                   FROM siswaa s
                   JOIN detail_prestasi dp ON s.id_siswa = dp.id_siswa
                   GROUP BY s.nama_siswa
                   ORDER BY jumlah DESC
                   LIMIT 1";
```

- **Fungsi:** `COUNT(dp.id_prestasi)` menghitung jumlah prestasi yang diraih oleh masing-masing siswa.
- **Relasi:** Tabel `siswa` dan `detail_prestasi` dihubungkan melalui `id_siswa`, yang memungkinkan penghitungan prestasi per siswa.
- **Pengelompokan dan Pengurutan:** `GROUP BY s.nama_siswa` mengelompokkan hasil berdasarkan nama siswa, dan `ORDER BY jumlah DESC` mengurutkan siswa berdasarkan jumlah prestasi dari yang terbanyak ke yang terendah. `LIMIT 1` digunakan untuk menampilkan hanya satu siswa dengan prestasi terbanyak.

Cara Kerja Fitur terhadap Agregasi

1. Ketika pengguna mengakses fitur ini, query agregasi dijalankan untuk mengambil data terkait prestasi siswa dari database.
2. **Pengelompokan dan Penghitungan:**
 - Pertama, jumlah prestasi dikelompokkan berdasarkan tingkatan, sehingga data terstruktur dan mudah dibaca.
 - Kedua, siswa yang memiliki prestasi terbanyak dihitung dan diurutkan untuk menampilkan siswa yang paling berprestasi.
3. **Penyajian Data:** Data hasil agregasi ditampilkan dalam format tabel di antarmuka pengguna. Ini memberikan gambaran yang jelas tentang prestasi siswa dan membantu dalam pengambilan keputusan oleh manajemen.

Autentikasi

- **Admin** → Memiliki akses penuh untuk mengelola data pengguna, siswa, guru, dan prestasi.

Analisis Program Autentikasi

```
<?php
session_start();
include "koneksi.php";
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $username = $_POST['username'];
    $password = md5($_POST['password']); // Hash password agar cocok dengan
    database
    $query = "SELECT * FROM users WHERE username='$username' AND
    password='$password'";
    $result = mysqli_query($koneksi, $query);
    if (mysqli_num_rows($result) == 1) {
        $_SESSION['username'] = $username;
        header("Location: index.php"); // Redirect ke halaman utama
    } else {
```

```
        echo "<script>alert('Username atau password salah!');  
window.location='login.php';</script>";  
    }  
}  
?>  
  
<?php  
session_start();  
include "koneksi.php";  
if (!isset($_SESSION['username'])) {  
    header("Location: login.php");  
    exit();  
}?>
```

1. Penerimaan Data dari Pengguna

Ketika pengguna mengisi formulir login dengan **username** dan **password**, sistem akan menerima data tersebut untuk diperiksa di database.

2. Enkripsi Password

Sebelum mencocokkan password yang dimasukkan dengan yang ada di database, sistem mengenkripsi password.

3. Pengecekan Data di Database

Setelah menerima username dan password, sistem akan mencari data tersebut di dalam database. Jika ada **kecocokan**, maka pengguna akan diberikan akses ke dalam sistem.

4. Pengalihan ke Halaman Utama

Setelah login berhasil, pengguna akan diarahkan ke halaman utama sistem agar bisa mulai menggunakan fitur yang tersedia