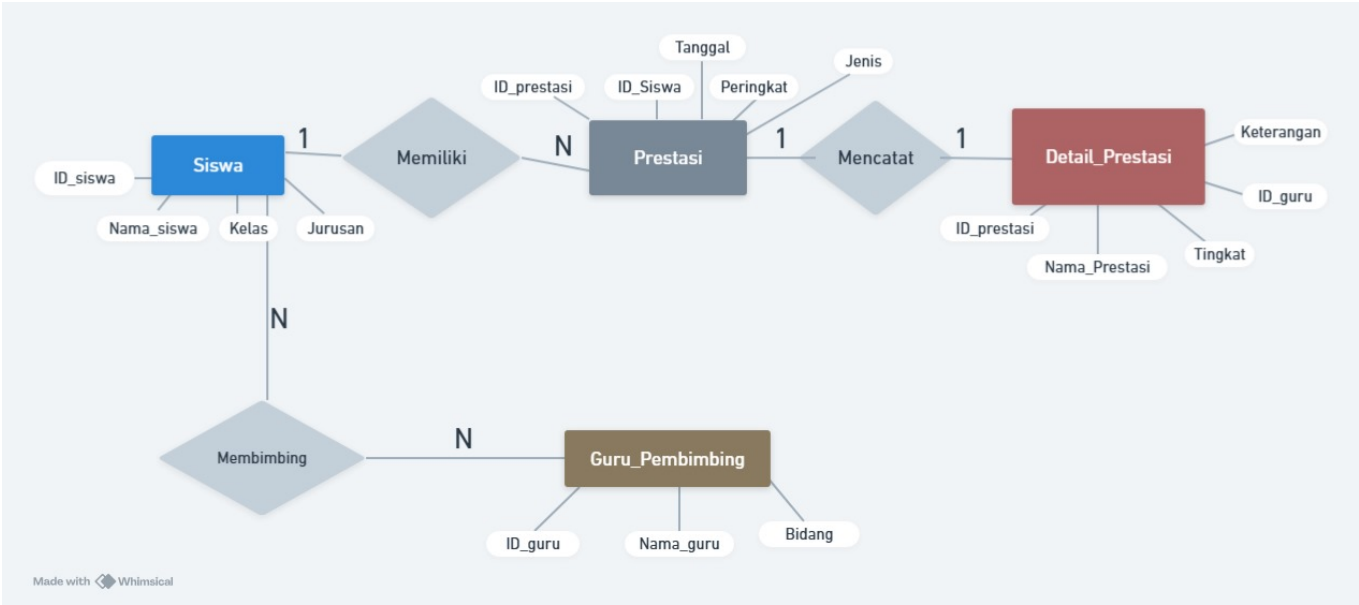


tabel Keaktifan

Tabel keaktifan Ujian Akhir

no	Nama kelompok	Skor	peran
1.	Mirsyah Emiyanti	2	Merangkum materi
2.	Siti Nurkhaerani	3	Merangkum materi dan mencari materi
3.	Jiliyan bintang k.	2	menuliskan materi
4.	Suci azizah	2	mencari materi

perancangan ERD



Data tabel keseluruhan  
tabel siswa

```
MariaDB [prestasi_siswa]> Select * from siswaa;
```

nis	nama	kelas	jurusan	jenis_kelamin
22312	Suci Azizah	XII RPL 2	RPL	perempuan
22318	Wisnu Sudrajat	XII AP 1	AP	laki-laki
22320	Fira	XII AP 1	AP	perempuan
22334	Mutiara	XII AK 2	AK	perempuan
22353	Syamsi SR	XII AP 2	AP	perempuan
22355	Condrado Alain S	XII RPL 2	RPL	laki-laki
22356	Adrian	XII RPL 2	RPL	laki-laki
22358	Clement Hermawan p	XII RPL 2	RPL	laki-laki
22359	Githania Agatha	XII RPL 2	RPL	perempuan
22360	Maisarah	XII AK 3	AK	perempuan
22363	Rian Mustary	XII PS 2	PS	laki-laki
22380	Nurul	XII AP 2	AP	perempuan
22412	Siti Nurkhaerani H	XII RPL 2	RPL	perempuan
23312	Gabriel	XI PS 2	PS	laki-laki
23314	Ahmad Givari	XI RPL	RPL	laki-laki
23315	Afrilia	XI RPL	RPL	perempuan

```
16 rows in set (0.000 sec)
```

tabel prestasi

```
MariaDB [prestasi_siswa]> Select * from prestasi;
```

id_prestasi	id_siswa	nama	tanggal_prestasi	kategori
1	22320	cerdas cermat ENIAC	2023-08-16	umum
2	22334	Matematika Akuntansi	2023-05-17	produktif
3	22359	Cerdas Cermat EIR	2023-12-20	produktif
4	23315	Lomba LKBB	2024-09-20	umum
5	22360	Cerdas Cermat PKN	2023-07-11	umum
6	22318	Lomba LKBB	2023-07-21	umum
7	22358	Lomba Desain	2022-11-14	produktif
8	22355	Cerdas Cermat EIR	2023-12-20	produktif
9	22380	Lomba LKBB	2023-07-21	umum
10	22355	Lomba Desain	2022-11-19	produktif
11	22355	Lomba LKBB	2023-07-21	umum
12	22320	Lomba Debat ENIAC	2023-08-16	umum

```
12 rows in set (0.001 sec)
```

tabel detail prestasi

```

MariaDB [prestasi_siswa]> SELECT * from detail_prestasi;
+-----+-----+-----+-----+-----+-----+
| id_prestasi | id_guru | nama_prestasi | tanggal_prestasi | kategori | peringkat |
+-----+-----+-----+-----+-----+-----+
| 1 | 1 | Cerdas Cermat ENIAC | 2023-08-16 | umum | 1 |
| 2 | 5 | Matematika Akuntansi | 2023-05-17 | produktif | 2 |
| 3 | 1 | Cerdas Cermat EIR | 2023-12-20 | produktif | 3 |
| 4 | 3 | Lomba LKBB | 2024-09-20 | umum | 2 |
| 5 | 4 | Cerdas Cermat PKN | 2023-07-11 | umum | 2 |
| 6 | 3 | Lomba LKBB | 2023-07-21 | umum | 4 |
| 7 | 2 | Lomba Desain EIR | 2022-11-14 | produktif | 2 |
| 8 | 1 | Cerdas Cermat EIR | 2023-12-20 | produktif | 3 |
| 9 | 3 | Lomba LKBB Delpas | 2023-07-21 | umum | 3 |
| 10 | 2 | Lomba Desain EIR | 2022-11-19 | produktif | 3 |
| 11 | 3 | Lomba LKBB Delpas | 2023-07-21 | umum | 3 |
| 12 | 1 | Lomba Debat ENIAC | 2023-08-16 | umum | 1 |
+-----+-----+-----+-----+-----+-----+
12 rows in set (0.004 sec)

```

tabel guru pembimbing

```

MariaDB [prestasi_siswa]> Select * from guru_pembimbing;
+-----+-----+-----+
| id_guru | nama_guru | bidang |
+-----+-----+-----+
| 1 | Ibrahim | IT |
| 2 | Fajar | desain |
| 3 | Fajar | lkbb |
| 4 | Anis | pkn |
| 5 | Tenrite | matematika |
+-----+-----+-----+
5 rows in set (0.000 sec)

```

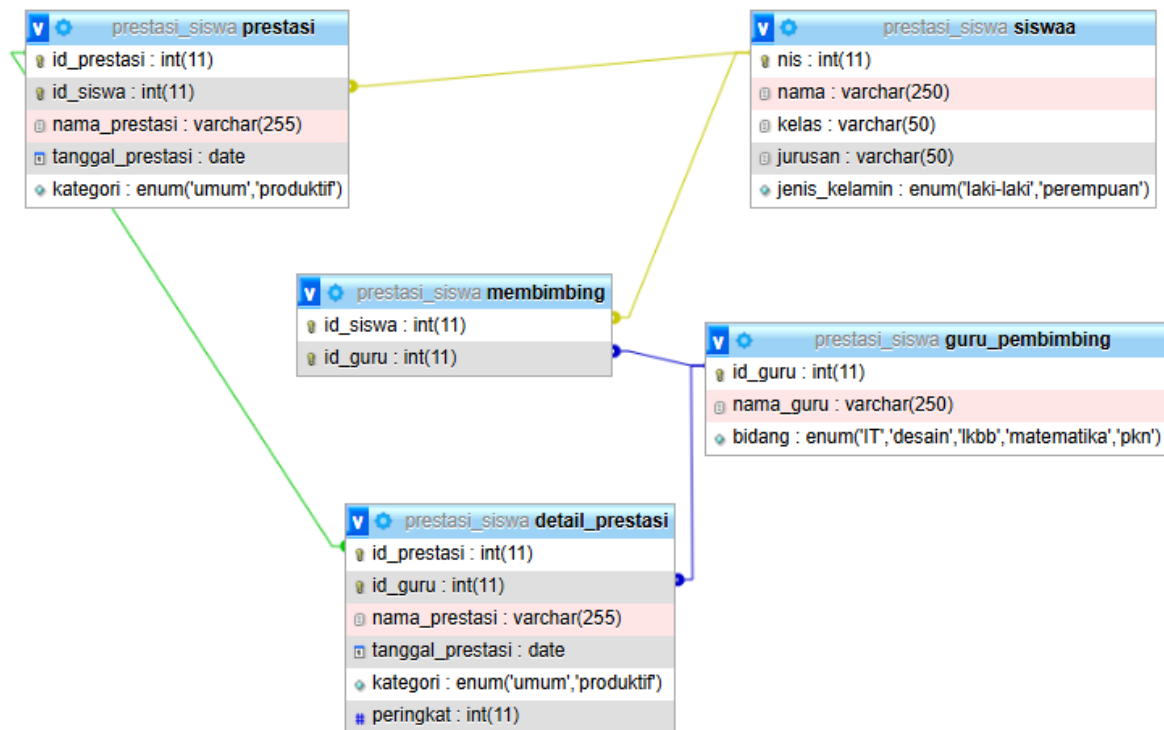
tabel membimbing

```

MariaDB [prestasi_siswa]> SELECT * from membimbing;
+-----+-----+
| id_siswa | id_guru |
+-----+-----+
| 22320 | 1 |
| 22355 | 1 |
| 22355 | 2 |
| 22355 | 3 |
| 22359 | 1 |
| 22360 | 5 |
+-----+-----+
5 rows in set (0.001 sec)

```

Bukti Relasi



## Penjelasan :

### 1. Tabel `siswaa`

#### 1. Kolom Utama:

- `nis` (Nomor Induk Siswa) sebagai **Primary Key**.

#### 2. Deskripsi:

- Tabel ini menyimpan data siswa, seperti nama, kelas, jurusan, dan jenis kelamin.

#### 3. Relasi:

- Tabel ini berelasi dengan tabel `prestasi` melalui kolom `nis = id_siswa`.
- Tabel ini juga berelasi dengan tabel `membimbing` melalui kolom `nis = id_siswa`.

### 2. Tabel `prestasi`

#### 1. Kolom Utama:

- `id_prestasi` sebagai **Primary Key**.

## 2. Deskripsi:

- Tabel ini menyimpan data tentang prestasi yang diraih siswa, seperti nama prestasi, tanggal prestasi, dan kategori prestasi (umum atau produktif).

## 3. Relasi:

- **Ke tabel `siswaa`:** Kolom `id_siswa` di tabel `prestasi` berelasi dengan kolom `nis` di tabel `siswaa`.
  - Relasi ini menunjukkan siapa siswa yang mendapatkan prestasi tertentu.
  - **Ke tabel `detail_prestasi`:** Kolom `id_prestasi` di tabel `prestasi` berelasi dengan kolom yang sama di tabel `detail_prestasi`.
- 

## 3. Tabel `guru_pembimbing`

### 1. Kolom Utama:

- `id_guru` sebagai **Primary Key**.

### 2. Deskripsi:

- Tabel ini menyimpan data guru pembimbing, termasuk nama guru dan bidang keahlian (seperti IT, desain, dll).

### 3. Relasi:

- **Ke tabel `membimbing`:** Kolom `id_guru` di tabel `guru_pembimbing` berelasi dengan kolom yang sama di tabel `membimbing`.

## 4. Tabel `membimbing`

### 1. Kolom Utama:

- Kombinasi `id_siswa` dan `id_guru` (komposit key).

### 2. Deskripsi:

- Tabel ini menjelaskan hubungan antara siswa dan guru, yaitu guru mana yang membimbing siswa tertentu.

### 3. Relasi:

- **Ke tabel `siswaa`:** Kolom `id_siswa` di tabel `membimbing` berelasi dengan kolom `nis` di tabel `siswaa`.
- **Ke tabel `guru_pembimbing`:** Kolom `id_guru` di tabel `membimbing` berelasi dengan kolom `id_guru` di tabel `guru_pembimbing`.

## 5. Tabel `detail_prestasi`

### 1. Kolom Utama:

- `id_prestasi` sebagai **Primary Key**.

### 2. Deskripsi:

- Tabel ini menyimpan detail lebih lanjut tentang prestasi siswa, seperti nama prestasi, tanggal, kategori, dan peringkat.

### 3. Relasi:

- **Ke tabel `prestasi`:** Kolom `id_prestasi` di tabel `detail_prestasi` berelasi dengan kolom yang sama di tabel `prestasi`.
- **Ke tabel `guru_pembimbing`:** Kolom `id_guru` di tabel `detail_prestasi` berelasi dengan kolom yang sama di tabel `guru_pembimbing`.
- Relasi ini menunjukkan guru pembimbing yang terlibat dalam membantu siswa mencapai prestasi tertentu.

## Relasi Utama

### 1. Siswa dan Prestasi:

- Relasi antara tabel `siswaa` dan `prestasi` menunjukkan siswa mana yang mendapatkan prestasi tertentu.

### 2. Guru dan Siswa:

- Relasi antara tabel `guru_pembimbing` dan `siswaa` melalui tabel `membimbing` menunjukkan hubungan guru-siswa (guru membimbing siswa tertentu).

### 3. Prestasi dan Guru:

- Relasi antara tabel `prestasi`, `detail_prestasi`, dan `guru_pembimbing` menunjukkan prestasi yang diraih siswa dengan bantuan guru pembimbing tertentu.

## Contoh penggunaan menggunakan code

### 1. menampilkan data guru dan prestasi tertinggi yang didapatkan siswa

code:

```
SELECT
-> pg.nama_guru, MAX(dp.peringkat) AS prestasi_tertinggi
-> FROM guru_pembimbing pg
-> JOIN detail_prestasi dp ON pg.id_guru = dp.id_guru
-> JOIN prestasi p ON dp.id_prestasi = p.id_prestasi
-> GROUP BY pg.id_guru;
```

Hasil :

```
MariaDB [prestasi_siswa]> SELECT
-> pg.nama_guru, MAX(dp.peringkat) AS prestasi_tertinggi
-> FROM guru_pembimbing pg
-> JOIN detail_prestasi dp ON pg.id_guru = dp.id_guru
-> JOIN prestasi p ON dp.id_prestasi = p.id_prestasi
-> GROUP BY pg.id_guru;
+-----+-----+
| nama_guru | prestasi_tertinggi |
+-----+-----+
| Ibrahim  | 3 |
| Fajar    | 3 |
| Fajar    | 4 |
| Anis     | 2 |
| Tenrite  | 2 |
+-----+-----+
5 rows in set (0.055 sec)
```

#### 1. `SELECT pg.nama_guru, MAX(p.peringkat) AS prestasi_tertinggi`

- **SELECT :**  
Code ini digunakan untuk menampilkan data yang sudah di masukkan
- **`pg.nama_guru`:**  
code ini memilih nama guru dari tabel `guru_pembimbing` untuk ditampilkan dalam hasil query.

- *MAX(p.peringkat):*

Fungsi agregasi MAX() digunakan untuk mencari nilai peringkat tertinggi dari tabel prestasi.

- *AS prestasi\_tertinggi:*

Memberi nama alias "prestasi\_tertinggi" untuk kolom hasil fungsi MAX() sehingga lebih mudah dipahami.

## 2. FROM guru\_pembimbing pg

- Menentukan tabel utama, yaitu guru\_pembimbing, dengan alias g.
- Alias ini digunakan untuk mempersingkat penulisan referensi tabel dalam query.

### \*3. JOIN detail\_prestasi d ON pg.id\_guru = dp.id\_guru

- **JOIN** menghubungkan tabel guru\_pembimbing (pg) dengan tabel detail\_prestasi (dp).
- Kondisi **ON** pg.id\_guru = dp.id\_guru berarti data guru akan dicocokkan berdasarkan kolom id\_guru pada kedua tabel.
- Relasi ini memastikan setiap data prestasi terkait dengan guru yang membimbingnya.

### 4. JOIN prestasi p ON dp.id\_prestasi = p.id\_prestasi

- **JOIN** menghubungkan tabel detail\_prestasi (dp) dengan tabel prestasi (p).
- Kondisi **ON** d.id\_prestasi = p.id\_prestasi memastikan data prestasi terkait dengan detail prestasi tertentu.

## 5. GROUP BY pg.id\_guru

- **GROUP BY** digunakan untuk mengelompokkan data berdasarkan kolom id\_guru (unik untuk setiap guru).
- Setiap kelompok data berisi semua baris yang berkaitan dengan satu guru.
- Setelah data dikelompokkan, fungsi agregasi seperti MAX() dapat bekerja pada setiap kelompok secara efektif.

## 2. Menampilkan data guru yang paling banyak membimbing siswa

code:



SELECT

```
-> pg.nama_guru, COUNT(m.id_siswa) AS total_siswa  
-> FROM guru_pembimbing pg  
-> JOIN membimbing m ON pg.id_guru = m.id_guru  
-> GROUP BY pg.id_guru  
-> HAVING total_siswa > 1;
```

hasil:

```
MariaDB [prestasi_siswa]> SELECT  
-> pg.nama_guru, COUNT(m.id_siswa) AS total_siswa  
-> FROM guru_pembimbing pg  
-> JOIN membimbing m ON pg.id_guru = m.id_guru  
-> GROUP BY pg.id_guru  
-> HAVING total_siswa > 1;  
+-----+-----+  
| nama_guru | total_siswa |  
+-----+-----+  
| Ibrahim   |          3 |  
+-----+-----+  
1 row in set (0.001 sec)
```

## 1. SELECT pg.nama\_guru, COUNT(m.id\_siswa) AS total\_siswa

- SELECT :  
Code ini digunakan untuk menampilkan data yang sudah di masukkan
- pg.nama\_guru:  
Bagian ini memilih nama guru dari tabel guru\_pembimbing untuk ditampilkan dalam hasil query.
- COUNT(m.id\_siswa):  
Fungsi agregasi COUNT() digunakan untuk menghitung jumlah siswa (id\_siswa) yang dibimbing oleh masing-masing guru.
- AS total\_siswa:  
Memberikan nama alias "total\_siswa" pada kolom hasil hitungan sehingga lebih mudah dibaca.

## 2. FROM guru\_pembimbing pg

- Menentukan tabel utama yang akan digunakan, yaitu guru\_pembimbing, dengan alias **pg**.
- Alias ini digunakan agar penulisan lebih singkat dan mudah saat mereferensikan tabel dalam query.

### 3. JOIN membimbing m ON pg.id\_guru = m.id\_guru

- **JOIN** menghubungkan tabel guru\_pembimbing (pg) dengan tabel membimbing (m).
- Kondisi **ON** `g.id_guru = m.id_guru`:
  - Menghubungkan baris di tabel guru\_pembimbing dengan baris di tabel membimbing berdasarkan kolom `id_guru`.
  - Artinya, query akan mencocokkan data setiap guru dengan data siswa yang dibimbing oleh guru tersebut.

### 4. GROUP BY pg.id\_guru

- **GROUP BY** digunakan untuk mengelompokkan data berdasarkan kolom `id_guru`.
- Setiap kelompok data mewakili satu guru, dan fungsi agregasi seperti **COUNT()** akan menghitung jumlah siswa dalam setiap kelompok.

### 5. HAVING total\_siswa > 1

- **HAVING** digunakan untuk memfilter hasil setelah data dikelompokkan.
- Kondisi `total_siswa > 1` berarti hanya guru yang membimbing lebih dari 1 siswa yang akan ditampilkan.
- Misalnya, jika ada guru yang hanya membimbing 1 siswa, data mereka akan dikeluarkan dari hasil.