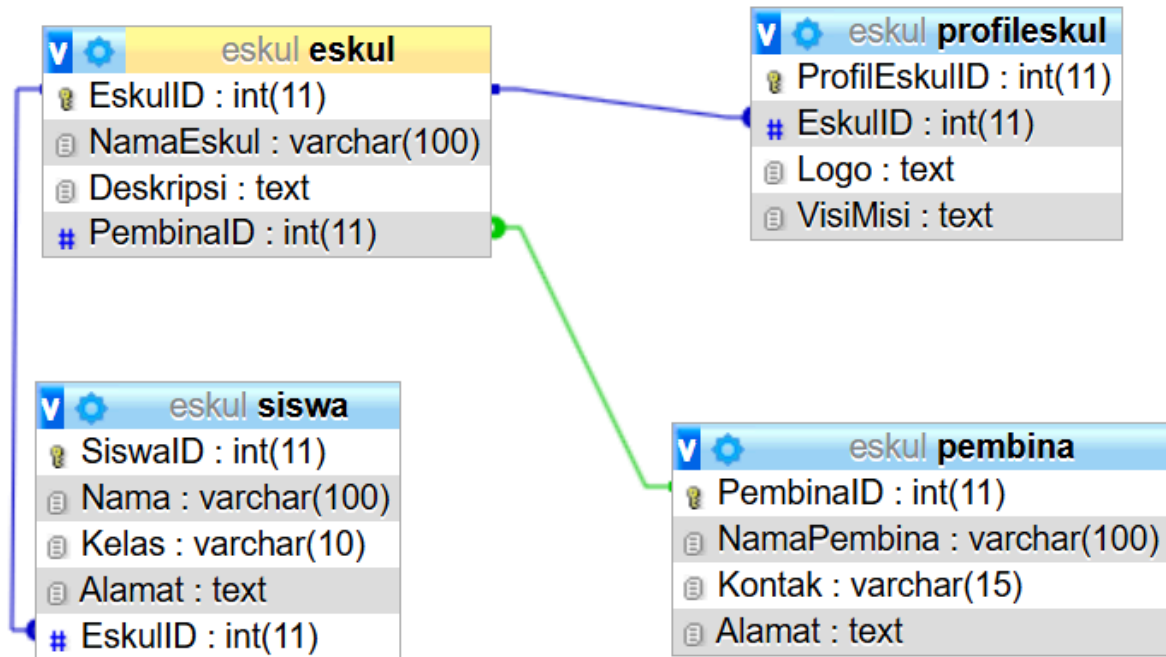


# Proyek Kelompok 5 Ujian Akhir

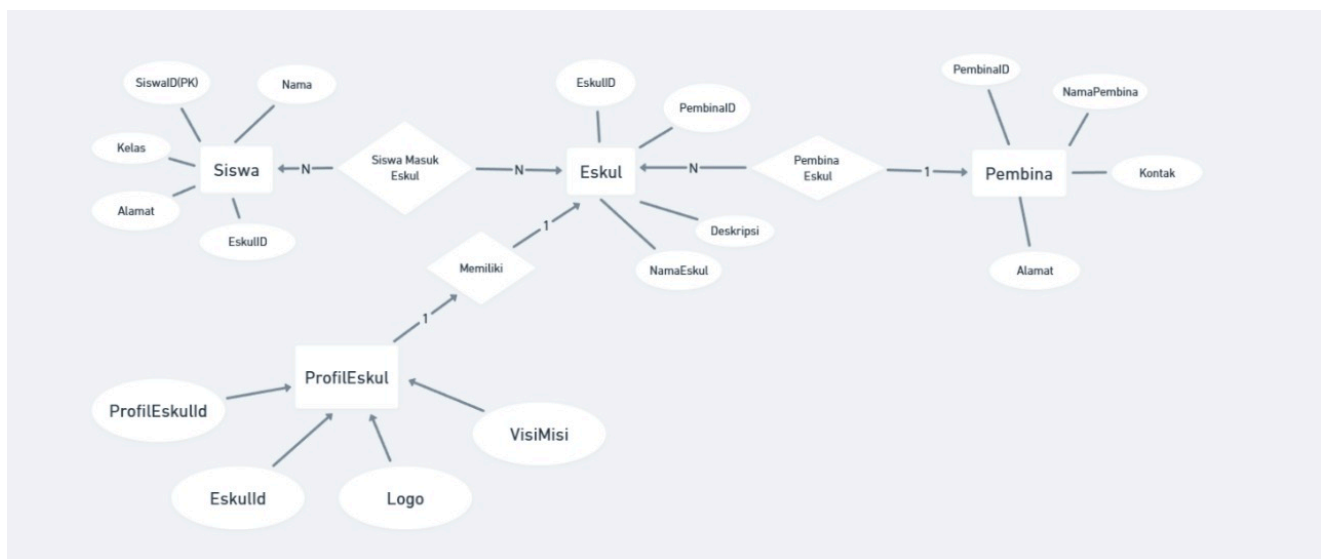
## SOAL 1

### Struktur ERD dan Database

Database :



Struktur ERD:



Untuk database di kelompok kami yaitu merancang database dan erd tentang eskul sekolah. di database ini terdiri dari 4 tabel yaitu tabel "eskul", tabel

"profileskul", tabel "siswa", dan tabel "pembina".

## Hubungan dan Kardinalitas:

### 1. Siswa - Eskul

- Kardinalitas: Banyak ke Banyak (Many-to-Many)
- Penjelasan: Seorang siswa(Siswa) dapat memiliki lebih dari satu baris data yang berhubungan dengan tabel eskul, dimana baris data yang berhubungan dengan eskul pun bisa lebih dari satu. Hal ini dikarenakan, **Siswa dapat memiliki lebih dari satu eskul**. Sebaiknya, hal ini di implementasikan dengan sebuah tabel pivot, misal dengan nama siswa\_eskul.

### 2. Eskul - Pembina

- Kardinalitas: Banyak ke Satu (Many-to-One)
- Penjelasan: Setiap Eskul memiliki satu Pembina (pembimbing), tetapi seorang Pembina bisa membimbing banyak Eskul. Oleh karena itu, baris data pembina pada tabel eskul dapat lebih dari satu, dan lebih dari satu baris data id\_pembina pada tabel eskul berhubungan dengan satu baris data pada tabel pembina, yaitu pada id\_pembina.

### 3. eskul - profileskul

- Kardinalitas: Banyak ke Banyak (One-to-One)
- Penjelasan: Sebuah Eskul (Tabel Eskul) hanya memiliki lebih satu baris data yang berhubungan dengan tabel profileskul, dimana baris data yang berhubungan dengan profileskul hanya dapat dihubungkan dengan satu baris data di tabel Eskul. Hal ini dikarenakan, **Sebuah eskul memiliki sebuah profil dan profil sebuah eskul tidak mungkin dimiliki oleh 2 eskul secara bersamaan**.

Berikut penjelasan dari masing-masing tabel:

## Tabel Eskul

### Struktur Query:

```
CREATE TABLE Eskul (  
  EskulID INT PRIMARY KEY,  
  NamaEskul VARCHAR(100),  
  Deskripsi TEXT,  
  PembinaID INT,  
  CONSTRAINT fk_PembinaID FOREIGN KEY (PembinaID) REFERENCES Pembina(PembinaID)  
  ON DELETE SET NULL );
```

hasil:

Field	Type	Null	Key	Default	Extra
EskulID	int(11)	NO	PRI	NULL	
NamaEskul	varchar(100)	YES		NULL	
Deskripsi	text	YES		NULL	
PembinaID	int(11)	YES	MUL	NULL	

4 rows in set (0.02 sec)

Tabel eskul adalah tabel utama yang berperan penting dalam database ini. Tabel ini berisi macam-macam eskul yang ada pada sekolah. Tabel ini terdiri dari 4 kolom yaitu kolom "EskulID", kolom "NamaEskul", kolom "Deskripsi", dan kolom "PembinaID". Untuk Foreign keynya, tabel ini berelasi dengan tabel "pembina", dimana tiap eskul memiliki pembina dan diinisialisasikan dengan "pembinaID" yang diambil dari tabel pembina.

### Penjelasan Query:

#### 1. CREATE TABLE Eskul

- **Fungsi:** Membuat tabel baru bernama `Eskul` yang akan digunakan untuk menyimpan data ekstrakurikuler.
- **Kolom-kolom yang didefinisikan dalam tabel:** `EskulID`, `NamaEskul`, `Deskripsi`, dan `PembinaID`.

#### 2. EskulID INT PRIMARY KEY

- **Kolom:** `EskulID`
  - Tipe data: **INT** (Integer), digunakan untuk menyimpan ID unik untuk setiap ekstrakurikuler.
  - **PRIMARY KEY:** Menandakan bahwa kolom ini adalah **Primary Key**, sehingga nilainya harus unik dan tidak boleh `NULL`.

#### 3. NamaEskul VARCHAR(100)

- **Kolom:** `NamaEskul`
  - Tipe data: **VARCHAR(100)**, digunakan untuk menyimpan nama ekstrakurikuler dalam bentuk teks dengan panjang maksimum 100 karakter.
  - Contoh: "Basket", "Seni", "Pramuka".

#### 4. Deskripsi TEXT

- **Kolom:** `Deskripsi`

- Tipe data: **TEXT**, digunakan untuk menyimpan deskripsi atau informasi tambahan tentang ekstrakurikuler.
- Tidak ada batasan panjang tertentu, tetapi disarankan tidak terlalu panjang.

## 5. PembinaID INT

- **Kolom:** `PembinaID`
  - Tipe data: **INT** (Integer), digunakan untuk menyimpan ID pembina.
  - **Relasi:** Kolom ini adalah **Foreign Key** yang merujuk ke kolom `PembinaID` di tabel **Pembina**.

## 6. CONSTRAINT fk\_PembinaID

- **Fungsi:** Membuat constraint atau batasan pada kolom `PembinaID` untuk menjadikannya **Foreign Key**.
- **Nama Constraint:** `fk_PembinaID`.
- **Foreign Key:**
  - Kolom `PembinaID` di tabel `Eskul` merujuk ke kolom `PembinaID` di tabel `Pembina`.
  - Relasi ini menunjukkan bahwa setiap ekstrakurikuler memiliki pembina yang terkait di tabel **Pembina**.

## 7. ON DELETE SET NULL

- **Fungsi:** Menentukan perilaku ketika data di tabel **Pembina** dihapus.
- **SET NULL:** Jika data pembina yang terkait dihapus dari tabel **Pembina**, nilai `PembinaID` di tabel **Eskul** akan diatur menjadi `NULL`.
  - **Alasan:** Ini memastikan data ekstrakurikuler tetap ada meskipun pembina yang terkait dihapus.

## Isi Data Tabel Eskul:

```
MariaDB [eskul]> Select * from eskul;
```

EskulID	NamaEskul	Deskripsi	PembinaID
1	Basket	Ekstrakurikuler olahraga basket	1
2	Seni	Ekstrakurikuler Sanggar Seni	NULL
3	Badminton	Ekstrakurikuler Bulu Tangkis	1

3 rows in set (0.01 sec)

## Tabel Pembina

### Struktur Query:

```
CREATE TABLE Pembina (
    PembinaID INT PRIMARY KEY,
    NamaPembina VARCHAR(100),
    Kontak VARCHAR(15),
    Alamat TEXT
);
```

Hasil:

```
MariaDB [eskul]> desc pembina;
```

Field	Type	Null	Key	Default	Extra
PembinaID	int(11)	NO	PRI	NULL	
NamaPembina	varchar(100)	YES		NULL	
Kontak	varchar(15)	YES		NULL	
Alamat	text	YES		NULL	

4 rows in set (0.04 sec)

Penjelasan Query:

### 1. CREATE TABLE Pembina

- **Fungsi:** Membuat tabel baru bernama `Pembina` yang akan digunakan untuk menyimpan data pembina ekstrakurikuler.
- **Kolom-kolom yang didefinisikan dalam tabel:** `PembinaID`, `NamaPembina`, `Kontak`, dan `Alamat`.

### 2. PembinaID INT PRIMARY KEY

- **Kolom:** `PembinaID`
  - Tipe data: **INT** (Integer), digunakan untuk menyimpan ID unik untuk setiap pembina.
  - **PRIMARY KEY:** Menandakan bahwa kolom ini adalah **Primary Key**, sehingga nilainya harus unik dan tidak boleh bernilai `NULL`.
  - Contoh: 1, 2, 3, ...

### 3. NamaPembina VARCHAR(100)

- **Kolom:** `NamaPembina`
  - Tipe data: **VARCHAR(100)**, digunakan untuk menyimpan nama pembina dengan panjang maksimum 100 karakter.
  - Contoh: "Pak Fajar", "Bu Hasnidar".

#### 4. Kontak VARCHAR(15)

- **Kolom:** `Kontak`
  - Tipe data: **VARCHAR(15)**, digunakan untuk menyimpan nomor kontak pembina dengan panjang maksimum 15 karakter.
  - Contoh: "081234567890".
  - Batasan panjang ini cukup untuk menyimpan format nomor telepon nasional atau internasional sederhana.

#### 5. Alamat TEXT

- **Kolom:** `Alamat`
  - Tipe data: **TEXT**, digunakan untuk menyimpan alamat pembina dalam bentuk teks.
  - Tidak memiliki batasan panjang tertentu.
  - Contoh: "Jl. Samalona Utara No. 16", "Jl. Sultan Alauddin No. 20".

#### Isi Data tabel Pembina

```
MariaDB [esku1]> SELECT * FROM pembina;
```

PembinaID	NamaPembina	Kontak	Alamat
1	Pak Fajar	081234567890	BTP
2	Pak Ebhi	081345678901	Antang

2 rows in set (0.00 sec)

### Tabel Siswa

#### Struktur Query:

```
CREATE TABLE Siswa (  
  SiswaID INT PRIMARY KEY,  
  Nama VARCHAR(100), Kelas VARCHAR(10),  
  Alamat TEXT, Esku1ID INT,  
  CONSTRAINT fk_Esku1ID FOREIGN KEY (Esku1ID) REFERENCES Esku1(Esku1ID) ON DELETE  
  SET NULL );
```

#### Hasil:

```
MariaDB [esku1]> desc siswa;
```

Field	Type	Null	Key	Default	Extra
SiswaID	int(11)	NO	PRI	NULL	
Nama	varchar(100)	YES		NULL	
Kelas	varchar(10)	YES		NULL	
Alamat	text	YES		NULL	
Esku1ID	int(11)	YES	MUL	NULL	

5 rows in set (0.02 sec)

## Penjelasan Query:

### 1. CREATE TABLE Siswa

- **Fungsi:** Membuat tabel baru bernama `Siswa` yang akan digunakan untuk menyimpan data siswa yang terdaftar dalam ekstrakurikuler.
- **Kolom-kolom yang didefinisikan dalam tabel:** `SiswaID`, `Nama`, `Kelas`, `Alamat`, dan `Esku1ID`.

### 2. SiswaID INT PRIMARY KEY

- **Kolom:** `SiswaID`
  - Tipe data: **INT** (Integer), digunakan untuk menyimpan ID unik untuk setiap siswa.
  - **PRIMARY KEY:** Menandakan bahwa kolom ini adalah **Primary Key**, sehingga nilainya harus unik dan tidak boleh bernilai `NULL`.
  - Contoh: 1, 2, 3, ...

### 3. Nama VARCHAR(100)

- **Kolom:** `Nama`
  - Tipe data: **VARCHAR(100)**, digunakan untuk menyimpan nama siswa dengan panjang maksimum 100 karakter.
  - Contoh: "Andi", "Budi", "Citra".

### 4. Kelas VARCHAR(10)

- **Kolom:** `Kelas`
  - Tipe data: **VARCHAR(10)**, digunakan untuk menyimpan informasi kelas siswa dengan panjang maksimum 10 karakter.
  - Contoh: "10-A", "11-B".

### 5. Alamat TEXT

- **Kolom:** `Alamat`
  - Tipe data: **TEXT**, digunakan untuk menyimpan alamat siswa dalam bentuk teks panjang.
  - Contoh: "Jl. Samalona Utara No. 16", "Jl. Sultan Alauddin No. 20".

## 6. EskulID INT

- **Kolom:** `EskulID`
  - Tipe data: **INT** (Integer), digunakan untuk menyimpan ID ekstrakurikuler (Foreign Key) yang diikuti siswa.
  - **Relasi:** Kolom ini menjadi **Foreign Key** yang merujuk ke kolom `EskulID` di tabel **Eskul**.
  - Contoh: Jika siswa mengikuti ekstrakurikuler **Basket**, maka `EskulID` akan mengacu pada **EskulID** dari ekstrakurikuler Basket di tabel **Eskul**.

## 7. CONSTRAINT fk\_EskulID FOREIGN KEY

- **Fungsi:** Membuat constraint (batasan) pada kolom `EskulID` untuk menjadikannya **Foreign Key**.
- **Nama Constraint:** `fk_EskulID`.
- **Foreign Key:**
  - Kolom `EskulID` di tabel **Siswa** merujuk ke kolom `EskulID` di tabel **\*\*Eskul`**.
  - Relasi ini menunjukkan bahwa setiap siswa bisa mengikuti salah satu ekstrakurikuler yang ada di tabel **\*\*Eskul`**.

## 8. ON DELETE SET NULL

- **Fungsi:** Menentukan perilaku ketika data di tabel **Eskul** yang dirujuk dihapus.
- **SET NULL:** Jika data di tabel **Eskul** dihapus, maka nilai `EskulID` di tabel **Siswa** akan diatur menjadi `NULL`.
  - **Alasan:** Ini memastikan bahwa data siswa tetap ada, meskipun data ekstrakurikuler yang mereka ikuti telah dihapus.

## Isi data tabel Siswa



```
MariaDB [esku1]> SELECT * FROM siswa;
```

SiswaID	Nama	Kelas	Alamat	Esku1ID
1	Condrado	10-A	Jl. Samalona Utara No. 16	2
2	Adrian	10-B	Jl. Pampang No. 11	1
3	Fahri	11-A	Jl. Cumi-cumi No. 30	3
4	Nabil	11-B	Jl. maccini No. 40	1

```
4 rows in set (0.00 sec)
```

## Tabel ProfilEsku1

### Struktur Query:

```
CREATE TABLE ProfilEsku1 (
  ProfilEsku1ID INT PRIMARY KEY,
  Esku1ID INT,
  Logo TEXT,
  VisiMisi TEXT,
  CONSTRAINT fk_ProfilEsku1_Esku1ID FOREIGN KEY (Esku1ID) REFERENCES
Esku1(Esku1ID) ON DELETE CASCADE
);
```

### Hasil:

```
MariaDB [esku1]> desc profilesku1
-> ;
```

Field	Type	Null	Key	Default	Extra
ProfilEsku1ID	int(11)	NO	PRI	NULL	
Esku1ID	int(11)	YES	MUL	NULL	
Logo	text	YES		NULL	
VisiMisi	text	YES		NULL	

```
4 rows in set (0.04 sec)
```

### Penjelasan:

#### 1. CREATE TABLE ProfilEsku1

- **Fungsi:** Membuat tabel baru bernama `ProfilEsku1` yang akan digunakan untuk menyimpan informasi tambahan mengenai ekstrakurikuler (esku1), seperti logo dan visi-misi.
- **Kolom-kolom yang didefinisikan dalam tabel:** `ProfilEsku1ID`, `Esku1ID`, `Logo`, dan `VisiMisi`.

## 2. ProfilEskulID INT PRIMARY KEY

- **Kolom:** `ProfilEskulID`
  - Tipe data: **INT** (Integer), digunakan untuk menyimpan ID unik untuk setiap profil eskul.
  - **PRIMARY KEY:** Menandakan bahwa kolom ini adalah **Primary Key**, sehingga nilainya harus unik dan tidak boleh bernilai `NULL`.
  - Contoh: 1, 2, 3, ...

## 3. EskulID INT

- **Kolom:** `EskulID`
  - Tipe data: **INT** (Integer), digunakan untuk menyimpan ID eskul yang dihubungkan dengan tabel `Eskul`.
  - **Relasi:** Kolom ini menjadi **Foreign Key** yang merujuk ke kolom `EskulID` di tabel `**Eskul`.
  - Contoh: Jika profil ini untuk eskul Basket, maka `EskulID` mengacu pada **EskulID** dari ekstrakurikuler Basket di tabel `**Eskul`.

## 4. Logo TEXT

- **Kolom:** `Logo`
  - Tipe data: **TEXT**, digunakan untuk menyimpan informasi terkait lokasi file atau URL gambar logo eskul.
  - Contoh: "basket\_logo.png", "[https://example.com/seni\\_logo.jpg](https://example.com/seni_logo.jpg)".

## 5. VisiMisi TEXT

- **Kolom:** `VisiMisi`
  - Tipe data: **TEXT**, digunakan untuk menyimpan visi dan misi dari setiap eskul.
  - Tidak ada batasan panjang tertentu.
  - Contoh: "Mengembangkan kreativitas seni siswa", "Melatih kemampuan kerja tim dan kepemimpinan".

## 6. CONSTRAINT fk\_ProfilEskul\_EskulID FOREIGN KEY

- **Fungsi:** Membuat constraint (batasan) pada kolom `EskulID` untuk menjadikannya **Foreign Key**.
- **Nama Constraint:** `fk_ProfilEskul_EskulID`.
- **Foreign Key:**
  - Kolom `EskulID` di tabel **ProfilEskul** merujuk ke kolom `EskulID` di tabel `**Eskul`.

- Relasi ini menunjukkan bahwa setiap profil hanya terkait dengan satu ekstrakurikuler.

## 7. ON DELETE CASCADE

- **Fungsi:** Menentukan perilaku ketika data di tabel **Eskul** yang dirujuk dihapus.
- **CASCADE** : Jika data di tabel **Eskul** dihapus, maka data yang terkait di tabel **ProfilEskul** juga akan dihapus secara otomatis.
  - **Alasan:** Hal ini memastikan bahwa tidak ada profil yang tertinggal di tabel **ProfilEskul** jika eskul yang bersangkutan sudah tidak ada.

### Isi Data tabel ProfilEskul:

```
MariaDB [esku1]> SELECT * FROM profilesku1;
```

ProfilEskulID	EskulID	Logo	VisiMisi
1	1	basket_logo.png	Mengasah kemampuan basket dan teamwork
2	2	seni_logo.png	Melatih keterampilan di bidang sanggar seni atau kesenian
3	3	badminton_logo.png	Melatih skil di bidang bulu tangkis

```
3 rows in set (0.01 sec)
```

## Soal 2

Pada soal kedua yaitu tampilkan datanya secara kontekstual dengan menggunakan query relasi, group by, dan having secara bersamaan dalam satu query. Buatlah minimal sebanyak 2 contoh.

### Query 1: Menampilkan Ekstrakurikuler dengan Jumlah Siswa >= 2

#### Struktur Query:

```
SELECT e>NamaEskul AS Nama_Eskul, COUNT(s.SiswaID) AS Jumlah_Siswa
FROM Siswa s
JOIN Eskul e ON s.EskulID = e.EskulID
GROUP BY e.EskulID
HAVING COUNT(s.SiswaID) >= 2;
```

#### Hasil:

```
MariaDB [eskul]> SELECT p>NamaPembina AS Nama_Pembina, COUNT(e.EskulID) AS Jumlah_Eskul
-> FROM Pembina p
-> JOIN Eskul e ON p.PembinaID = e.PembinaID
-> GROUP BY p.PembinaID
-> HAVING COUNT(e.EskulID) >= 1;
```

Nama_Pembina	Jumlah_Eskul
Pak Fajar	2

1 row in set (0.00 sec)

## Penjelasan:

- Dari data di tabel **Siswa**, siswa terhubung dengan ekstrakurikuler:
  - EskulID = 1 (Basket)**: Adrian dan Nabil → 2 siswa.
  - EskulID = 2 (Seni)**: Condrado → 1 siswa.
  - EskulID = 3 (Badminton)**: Fahri → 1 siswa.
- Hanya **Basket (EskulID = 1)** yang memiliki jumlah siswa **>= 2**.

## Analisis Struktur Query:

### 1. SELECT e>NamaEskul AS Nama\_Eskul, COUNT(s.SiswaID) AS Jumlah\_Siswa

- Fungsi:**
  - Memilih nama ekstrakurikuler dari tabel **Eskul** dan menghitung jumlah siswa yang terdaftar pada setiap eskul.
- Komponen:**
  - e>NamaEskul**: Memilih kolom **NamaEskul** dari tabel **Eskul** untuk mendapatkan nama ekstrakurikuler.
  - COUNT(s.SiswaID)**: Fungsi agregat untuk menghitung jumlah siswa yang terhubung ke setiap eskul berdasarkan kolom **SiswaID**.
  - AS**: Memberi alias pada kolom hasil, sehingga kolom **NamaEskul** diberi nama **Nama\_Eskul**, dan hasil **COUNT(s.SiswaID)** diberi nama **Jumlah\_Siswa**.

### 2. FROM Siswa s

- Fungsi:**
  - Menentukan tabel sumber data, yaitu tabel **Siswa**.
- Alias:**
  - Tabel **Siswa** diberi alias **s** untuk mempermudah penulisan dalam query.

### 3. JOIN Eskul e ON s.EskulID = e.EskulID

- Fungsi:**

- Menggabungkan tabel `Siswa` dengan tabel `Eskul` menggunakan **INNER JOIN**.
- Kondisi penggabungan adalah `s.EskulID = e.EskulID`, yang memastikan bahwa siswa hanya dihitung jika mereka terhubung ke sebuah eskul di tabel `Eskul`.

#### 4. GROUP BY e.EskulID

- **Fungsi:**
  - Mengelompokkan data berdasarkan kolom `EskulID` dari tabel `Eskul`.
  - Setiap grup akan merepresentasikan satu ekstrakurikuler.
  - Setelah data dikelompokkan, fungsi agregat seperti **COUNT(s.SiswaID)** dapat digunakan untuk menghitung jumlah siswa per grup.

#### 5. HAVING COUNT(s.SiswaID) >= 2

- **Fungsi:**
  - Memfilter grup hasil pengelompokan dengan kondisi bahwa hanya grup yang memiliki **jumlah siswa >= 2** yang akan ditampilkan.
- **Perbedaan dengan WHERE:**
  - `WHERE` digunakan untuk memfilter data sebelum pengelompokan.
  - `HAVING` digunakan untuk memfilter grup setelah fungsi agregat seperti **COUNT** diterapkan.

## Query 2: Menampilkan Pembina dengan Jumlah Ekstrakurikuler >= 1

### Struktur Query:

```
SELECT p>NamaPembina AS Nama_Pembina, COUNT(e.EskulID) AS Jumlah_Eskul
FROM Pembina p
JOIN Eskul e ON p.PembinaID = e.PembinaID
GROUP BY p.PembinaID
HAVING COUNT(e.EskulID) ≥ 1;
```

### Hasil:

```
+-----+-----+
| Nama_Pembina | Jumlah_Eskul |
+-----+-----+
| Pak Fajar    | 2            |
+-----+-----+
1 row in set (0.00 sec)
```

## Penjelasan:

- Dari data di tabel **Eskul** dan **Pembina**, relasi antara pembina dan eskul adalah:
  - **Pak Fajar (PembinaID = 1)** membimbing:
    - **Basket (EskulID = 1)**.
    - **Badminton (EskulID = 3)**.
  - **Pak Ebhi (PembinaID = 2)** membimbing:
    - **Seni (EskulID = 2)**.
- Pak Fajar membimbing **2 eskul**, sedangkan Pak Ebhi membimbing **1 eskul**.

## Analisis Query:

### 1. SELECT p>NamaPembina AS Nama\_Pembina, COUNT(e.EskulID) AS Jumlah\_Eskul

- **Fungsi:**
  - Memilih nama pembina dari tabel **Pembina** dan menghitung jumlah ekstrakurikuler yang mereka bimbing.
- **Komponen:**
  - **p>NamaPembina**: Mengambil kolom **NamaPembina** dari tabel **Pembina**, yang berisi nama pembina.
  - **AS Nama\_Pembina**: Memberi alias **Nama\_Pembina** untuk kolom **NamaPembina**, sehingga lebih mudah dibaca dalam hasil query.
  - **COUNT(e.EskulID)**: Menggunakan fungsi agregat **COUNT** untuk menghitung jumlah ekstrakurikuler (**EskulID**) yang dibimbing oleh masing-masing pembina.
  - **AS Jumlah\_Eskul**: Memberi alias **Jumlah\_Eskul** untuk hasil fungsi **COUNT(e.EskulID)**.

### 2. FROM Pembina p

- **Fungsi:**
  - Menentukan tabel sumber data, yaitu tabel **Pembina**.
- **Alias:**
  - Tabel **Pembina** diberi alias **p** untuk mempermudah penulisan query.

### 3. JOIN Eskul e ON p.PembinaID = e.PembinaID

- **Fungsi:**
  - Menggabungkan tabel **Pembina** dengan tabel **Eskul** menggunakan **INNER JOIN**.
  - Kondisi penggabungan adalah **ON p.PembinaID = e.PembinaID**, yang memastikan hanya data yang memiliki relasi antara tabel **Pembina**

dan tabel **Eskul** yang ditampilkan.

- **Hasil:**

- Setiap pembina akan digabungkan dengan ekstrakurikuler yang mereka bimbing.

#### 4. GROUP BY p.PembinaID

- **Fungsi:**

- Mengelompokkan hasil query berdasarkan kolom **PembinaID**.
- Setelah data dikelompokkan, fungsi agregat seperti **COUNT(e.EskulID)** dapat digunakan untuk menghitung jumlah ekstrakurikuler per pembina.

- **Catatan:**

- Kolom **p>NamaPembina** boleh digunakan dalam hasil tanpa pengelompokan eksplisit karena ini bergantung langsung pada **PembinaID** (Primary Key).

#### 5. HAVING COUNT(e.EskulID) >= 1

- **Fungsi:**

- Memfilter grup hasil pengelompokan sehingga hanya pembina yang membimbing **minimal 1 ekstrakurikuler** yang akan ditampilkan.

- **Perbedaan dengan WHERE:**

- **WHERE** digunakan untuk memfilter data sebelum pengelompokan.
- **HAVING** digunakan untuk memfilter grup setelah pengelompokan dan penerapan fungsi agregat seperti **COUNT**.

#### Tabel Keaktifan:

No	Nama	Nilai	Tugas
1.	Condrado	3	Membuat Catatan
2.	Adrian	3	Membantu Membuat Catatan
3.	Fahri Ilham	3	Membantu membuat Catatan dan mengurus upload file ke drive
4.	Muh. Nabil Maulana	1	Tidak Ada
5.	Yeremia Tasik	3	Membantu membuat catatan