

## **POST TEST 6 - SQL AGGREGATE■Jawaban (Queries) dan Petunjuk Screenshot**

File sumber soal: Soal PT6\_B.docx. Jawaban berikut berisi query SQL yang dapat langsung dijalankan pada MySQL/phpMyAdmin setelah Anda mengimpor file PT6\_GrupB.sql dari LMS. Ganti PT6\_NIM pada perintah CREATE DATABASE dengan NIM Anda (contoh: PT6\_71240808).

Catatan penting:

- Saya tidak memiliki file PT6\_GrupB.sql/schema di lingkungan ini, jadi nama tabel/kolom yang ada di bawah menggunakan nama umum yang sering dipakai (mis. studio, film, tiket, genre).
- Jika nama tabel/kolom di PT6\_GrupB.sql berbeda, sesuaikan nama tabel/kolom pada query berikut agar sesuai dengan skema yang Anda import.

Instruksi untuk screenshot (phpMyAdmin):

- 1) Pastikan database yang aktif adalah PT6\_<NIM> di sidebar phpMyAdmin.
- 2) Jalankan query pada tab SQL.
- 3) Ambil screenshot yang memperlihatkan: nama database di kiri, pesan 'SQL query executed' (atau similar), query yang Anda jalankan, dan hasil tabel (lihat CONTOH Pengerjaan di soal).

### **Soal 1 — Tampilkan nama studio dan total penjualan tiket dari semua film yang diproduksi.**

Catatan penamaan kolom yang diminta: 'studio' dan 'total\_penjualan'.

Query (sesuaikan nama tabel/kolom jika berbeda pada PT6\_GrupB.sql):

```
CREATE DATABASE PT6_<NIM>;
USE PT6_<NIM>;
SELECT s.nama_studio AS studio,
       COALESCE(SUM(t.jumlah_tiket), 0) AS total_penjualan
FROM studio s
JOIN film f ON s.studio_id = f.studio_id
LEFT JOIN tiket t ON f.film_id = t.film_id
GROUP BY s.nama_studio
ORDER BY total_penjualan DESC;
```

Screenshot: jalankan query di phpMyAdmin dan ambil SS yang memperlihatkan nama DB, pesan eksekusi, query, dan hasil.

### **Soal 2 — Tampilkan setiap genre beserta rata-rata durasi film-nya.**

Penamaan kolom yang diminta: 'genre' dan 'avg\_durasi'.

Query (sesuaikan nama tabel/kolom jika berbeda):

```
SELECT g.nama_genre AS genre,
       ROUND(AVG(f.durasi), 2) AS avg_durasi
FROM genre g
JOIN film f ON g.genre_id = f.genre_id
GROUP BY g.nama_genre
ORDER BY avg_durasi ASC;
```

Screenshot: jalankan query di phpMyAdmin dan ambil SS sesuai ketentuan.

### **Soal 3 — Tampilkan tahun rilis dan jumlah film yang dirilis pada tahun tersebut.**

Penamaan kolom yang diminta: 'tahun\_rilis' dan 'jumlah\_film'.

Query (sesuaikan nama tabel/kolom jika berbeda):

```
SELECT YEAR(f.tanggal_rilis) AS tahun_rilis,
       COUNT(*) AS jumlah_film
FROM film f
GROUP BY YEAR(f.tanggal_rilis)
ORDER BY tahun_rilis DESC;
```

Catatan: jika kolom sudah bertipe YEAR atau bernama 'tahun', gunakan kolom tersebut langsung.

Screenshot: jalankan dan ambil SS.