# SISTEM PERPUSTAKAAN SEKOLAH

# 1. Tabel Buku

ID Buku	Judul Buku	Penulis	Kategori	Stok
1	Algoritma dan Pemrograman	Andi Wijaya	Teknologi	5
2	Dasar-dasar Database	Budi Santoso	Teknologi	7
3	Matematika Diskrit	Rina Sari	Matematika	4
4	Sejarah Dunia	John Smith	Sejarah	3
5	Pemrograman Web dengan PHP	Eko Prasetyo	Teknologi	8
6	Sistem Operasi	Dian Kurniawan	Teknologi	6
7	Jaringan Komputer	Ahmad Fauzi	Teknologi	5
8	Cerita Rakyat Nusantara	Lestari Dewi	Sastra	9
9	Bahasa Inggris untuk Pemula	Jane Doe	Bahasa	10
10	Biologi Dasar	Budi Rahman	Sains	7
11	Kimia Organik	Siti Aminah	Sains	5
12	Teknik Elektro	Ridwan Hakim	Teknik	6
13	Fisika Modern	Albert Einstein	Sains	4
14	Manajemen Waktu	Steven Covey	Pengembangan	8
15	Strategi Belajar Efektif	Tony Buzan	Pendidikan	6

# 2. Tabel Siswa

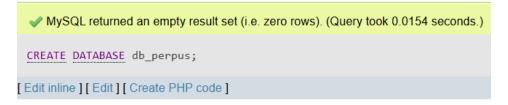
ID Siswa	Nama	Kelas	
1	Andi Saputra	X-RPL	
2	Budi Wijaya	X-TKJ	
3	Citra Lestari	XI-RPL	
4	Dewi Kurniawan	XI-TKJ	
5	Eko Prasetyo	XII-RPL	
6	Farhan Maulana	XII-TKJ	
7	Gita Permata	X-RPL	
8	Hadi Sucipto	X-TKJ	
9	Intan Permadi	XI-RPL	
10	Joko Santoso	XI-TKJ	
11	Kartika Sari	XII-RPL	
12	Lintang Putri	XII-TKJ	
13	Muhammad Rizky	X-RPL	
14	Novi Andriana	X-TKJ	
15	Olivia Hernanda	XI-RPL	

# 3. Tabel Peminjaman

ID Peminjaman	ID Siswa	ID Buku	Tanggal Pinjam	Tanggal Kembali	Status
1	11	2	2025-02-01	2025-02-08	Dipinjam
2	2	5	2025-01-28	2025-02-04	Dikembalikan
3	3	8	2025-02-02	2025-02-09	Dipinjam
4	4	10	2025-01-30	2025-02-06	Dikembalikan
5	5	3	2025-01-25	2025-02-01	Dikembalikan
6	15	7	2025-02-01	2025-02-08	Dipinjam
7	7	1	2025-01-29	2025-02-05	Dikembalikan
8	8	9	2025-02-03	2025-02-10	Dipinjam
9	13	4	2025-01-27	2025-02-03	Dikembalikan
10	10	11	2025-02-01	2025-02-08	Dipinjam

### **TUGAS**

1. Buatlah database dengan nama db perpus.



2. Buatlah table buku, siswa dan peminjaman.

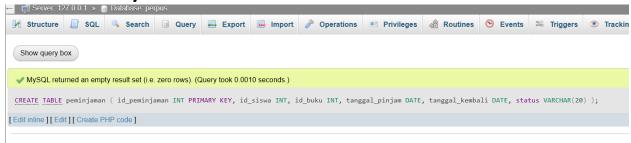
## Table Buku:



### Table siswa;



**Table Peminjaman:** 

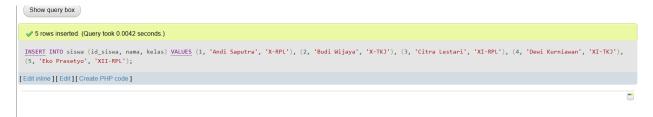


3. Input 5 record di setiap table menggunakan query INSERT, UPDATE, DELETE.

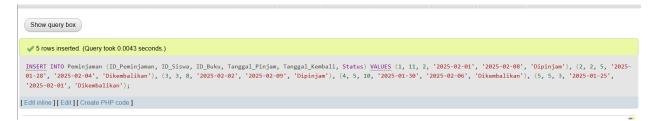
#### **Insert Buku:**



#### **Insert Siswa:**



### **Insert Peminjaman:**



4. Input 10 record di setiap table menggunakan stored procedure INSERT.

#### Insert buku:

```
✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0178 seconds.)

CREATE PROCEDURE insert_buku( IN p_id_buku INT, IN p_judul_buku VARCHAR(100), IN p_penulis VARCHAR(50), IN p_kategori VARCHAR(50), IN p_stok INT ) BEGIN INSERT INTO buku (id_buku, judul_buku, penulis, kategori, stok) VALUES (p_id_buku, p_judul_buku, p_penulis, p_kategori, p_stok); END;

Edit inline] [ Edit] [ Create PHP code ]
```

#### Insert Siswa:

```
✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0162 seconds.)

CREATE PROCEDURE insert_siswa( IN p_id_siswa INT, IN p_nama VARCHAR(50), IN p_kelas VARCHAR(10)) BEGIN INSERT INTO siswa (id_siswa, nama, kelas) VALUES (p_id_siswa, p_nama, p_kelas); END;

[Edit inline] [Edit] [Create PHP code]
```

## **Insert Peminjaman:**

```
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0132 seconds.)
```

CREATE PROCEDURE insert\_peminjaman( IN p\_id\_peminjaman INT, IN p\_id\_siswa INT, IN p\_id\_buku INT, IN p\_tanggal\_pinjam DATE, IN p\_tanggal\_kembali DATE, IN p\_status VARCHAR(20) 8 BGIN INSERT INTO Peminjaman (id\_peminjaman, id\_siswa, id\_buku, tanggal\_pinjam, tanggal\_kembali, status) VALUES (p\_id\_peminjaman, p\_id\_siswa, p\_id\_buku, p\_tanggal\_pinjam, p\_tanggal\_kembali, p\_status); END;

[ Edit inline ] [ Edit ] [ Create PHP code ]

# Memanggil (Buku)

```
CALL insert_buku(16, 'Struktur Data', 'Sarwono', 'Teknologi', 9);

CALL insert_buku(17, 'Kalkulus', 'Thomas', 'Matematika', 6);

CALL insert_buku(18, 'Sejarah Indonesia', 'Marwati', 'Sejarah', 5);

CALL insert_buku(19, 'Pemrograman Python', 'Mark Lutz', 'Teknologi', 12);

CALL insert_buku(20, 'Sistem Basis Data', 'C.J. Date', 'Teknologi', 7);

CALL insert_buku(21, 'Jaringan Komputer', 'Andrew Tanenbaum', 'Teknologi', 8);

CALL insert_buku(22, 'Sastra Indonesia', 'Pramoedya Ananta Toer', 'Sastra', 4);

CALL insert_buku(23, 'Bahasa Inggris Bisnis', 'Murphy', 'Bahasa', 11);

CALL insert_buku(24, 'Fisika Dasar', 'Halliday Resnick', 'Sains', 10);

CALL insert_buku(25, 'Kimia Analisis', 'Basset', 'Sains', 5);
```

## Memanggil (Siswa)

```
CALL insert_siswa(16, 'Rahmat Hidayat', 'X-RPL');

CALL insert_siswa(17, 'Siti Aisyah', 'X-TKJ');

CALL insert_siswa(18, 'Dimas Pratama', 'XI-RPL');

CALL insert_siswa(19, 'Rina Setiani', 'XI-TKJ');

CALL insert_siswa(20, 'Fahmi Nugraha', 'XII-RPL');

CALL insert_siswa(21, 'Lestari Dewi', 'XII-TKJ');

CALL insert_siswa(22, 'Bayu Saputra', 'X-RPL');

CALL insert_siswa(23, 'Indah Puspita', 'X-TKJ');

CALL insert_siswa(24, 'Rizal Fadillah', 'XI-RPL');

CALL insert_siswa(25, 'Putri Amelia', 'XI-TKJ');
```

```
Memanggil (Peminjaman)

CALL insert_peminjaman(1, 11, 2, '2025-02-01', '2025-02-08', 'Dipinjam');

CALL insert_peminjaman(2, 2, 5, '2025-01-28', '2025-02-04', 'Dikembalikan');

CALL insert_peminjaman(3, 3, 8, '2025-02-02', '2025-02-09', 'Dipinjam');

CALL insert_peminjaman(4, 4, 10, '2025-01-30', '2025-02-06', 'Dikembalikan');

CALL insert_peminjaman(5, 5, 3, '2025-01-25', '2025-02-01', 'Dikembalikan');

CALL insert_peminjaman(6, 15, 7, '2025-02-01', '2025-02-08', 'Dipinjam');

CALL insert_peminjaman(7, 1, 7, '2025-01-29', '2025-02-05', 'Dikembalikan');

CALL insert_peminjaman(8, 8, 9, '2025-02-03', '2025-02-01', 'Dipinjam');

CALL insert_peminjaman(9, 13, 4, '2025-01-27', '2025-02-03', 'Dikembalikan');

CALL insert_peminjaman(10, 10, 11, '2025-02-01', '2025-02-08', 'Dipinjam');

5. Buatlah stored procedure UPDATE, DELETE di setiap table.

Table Siswa:

Update

DELIMITER //
```

```
CREATE PROCEDURE update siswa(
```

```
IN p_id_siswa INT,
IN p_nama VARCHAR(50),
IN p_kelas VARCHAR(50)
)

BEGIN

UPDATE siswa

SET nama = p_nama,
kelas = p_kelas

WHERE id_siswa = p_id_siswa;

END //
```

**DELIMITER**;

```
Delete
DELIMITER //
CREATE PROCEDURE delete_siswa(
  IN p_id_siswa INT
)
BEGIN
  DELETE FROM siswa WHERE id_siswa = p_id_siswa;
END //
DELIMITER;
Table Buku:
Update
DELIMITER //
CREATE PROCEDURE update_buku(
 IN p_id_buku INT,
  IN p_judul VARCHAR(100),
 IN p_penulis VARCHAR(100),
 IN p_kategori VARCHAR(50),
 IN p_stok INT
)
BEGIN
  UPDATE buku
  SET judul = p_judul,
    penulis = p_penulis,
    kategori = p_kategori,
    stok = p_stok
 WHERE id_buku = p_id_buku;
```

```
END //
DELIMITER;
Delete
DELIMITER //
CREATE PROCEDURE delete_buku(
 IN p_id_buku INT
)
BEGIN
  DELETE FROM buku WHERE id_buku = p_id_buku;
END //
DELIMITER;
Table Peminjaman:
Update
DELIMITER //
CREATE PROCEDURE update_peminjaman(
  IN p_id_peminjaman INT,
  IN p_id_siswa INT,
  IN p_id_buku INT,
  IN p_tanggal_pinjam DATE,
  IN p_tanggal_kembali DATE,
 IN p_status ENUM('Dipinjam', 'Dikembalikan')
)
BEGIN
  UPDATE peminjaman
  SET id_siswa = p_id_siswa,
```

```
id_buku = p_id_buku,
    tanggal_pinjam = p_tanggal_pinjam,
    tanggal_kembali = p_tanggal_kembali,
    status = p_status
  WHERE id_peminjaman = p_id_peminjaman;
END //
DELIMITER;
Delete
DELIMITER //
CREATE PROCEDURE delete peminjaman(
  IN p_id_peminjaman INT
)
BEGIN
  DELETE FROM peminjaman WHERE id_peminjaman = p_id_peminjaman;
END //
DELIMITER;
6. Buatlah stored procedure untuk menampilkan seluruh record di setiap table.
Table Siswa
DELIMITER //
CREATE PROCEDURE get_all_siswa()
BEGIN
  SELECT * FROM siswa;
END //
DELIMITER;
```

```
Table Buku
DELIMITER //
CREATE PROCEDURE get_all_buku()
BEGIN
 SELECT * FROM buku;
END //
DELIMITER;
Table Peminjaman
DELIMITER //
CREATE PROCEDURE get_all_peminjaman()
BEGIN
  SELECT * FROM peminjaman;
END //
DELIMITER;
7. Stok buku pada saat dipinjam berkurang secara otamatis.
DELIMITER //
CREATE TRIGGER KurangiStokBuku
AFTER INSERT ON peminjaman
FOR EACH ROW
BEGIN
UPDATE buku
SET Stok = Stok - 1
WHERE Id_Buku = NEW.id_buku AND NEW.status='Dipinjam';
END //
```

```
8. Stok buku pada saat dikembalikan bertambah secara otomatis.
    DELIMITER //
   CREATE TRIGGER stok bertambah
   AFTER UPDATE ON peminjaman
   FOR EACH ROW
    BEGIN
      IF OLD.status = 'Dipinjam' AND NEW.status = 'Dikembalikan' THEN
        UPDATE buku
        SET stok = stok + 1
        WHERE id buku = NEW.id buku;
     END IF;
   END //
    DELIMITER;
9. Buatlah stored procedure untuk mengembalikan buku dan gunakan tanggal pengembalian
    sesuai dengan tanggal saat mengembalikan (CURRENT DATE).
DELIMITER //
CREATE PROCEDURE KembalikanBuku(
IN p_id_peminjaman INT
)
BEGIN
UPDATE peminjaman
SET TanggalKembali = CURRENT_DATE, status = 'Dikembalikan'
WHERE Id Peminjaman = p id peminjaman;
END //
10. Buatlah stored procedure untuk menampilkan daftar siswa yang pernah meminjam buku.
    DELIMITER //
   CREATE PROCEDURE GetSiswaPernahPinjam()
      SELECT DISTINCT s.id siswa, s.nama, s.kelas
     FROM siswa s
     JOIN peminjaman p ON s.id siswa = p.id siswa;
    END //
```

**DELIMITER**;

11. Buatlah stored procedure untuk menampilkan semua siswa, termasuk yang tidak pernah meminjam buku.

```
DELIMITER //
   CREATE PROCEDURE showAllSiswa()
    BEGIN
      SELECT s.id siswa, s.nama, s.kelas,
         IF(p.id_siswa IS NULL, 'Belum Pernah Meminjam', 'Pernah Meminjam') AS
   status peminjaman
      FROM siswa s
     LEFT JOIN peminjaman p ON s.id siswa = p.id siswa
      GROUP BY s.id siswa, s.nama, s.kelas;
    END //
    DELIMITER;
12. Buatlah stored procedure untuk menampilkan semua buku, termasuk yang belum pernah
    dipinjam.
DELIMITER //
CREATE PROCEDURE showAllBuku()
BEGIN
 SELECT b.id_buku, b.judul, b.pengarang, b.kategori, b.stok,
     IF(p.id buku IS NULL, 'Belum Pernah Dipinjam', 'Pernah Dipinjam') AS status peminjaman
 FROM buku b
 LEFT JOIN peminjaman p ON b.id buku = p.id buku
 GROUP BY b.id buku, b.judul, b.pengarang, b.kategori, b.stok;
END //
DELIMITER;
```

<sup>~</sup> PUSH File SQL ke gitHub dengan nama repository Database-Pepustakaan-Sekolah ~