

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-	2025-02-	Dipinjam

			01	08	
--	--	--	----	----	--

TUGAS

1. Buatlah database dengan nama db_perpus.

```
CREATE DATABASE perpustakaan;
```

2. Buatlah table buku, siswa dan peminjaman.

```
CREATE TABLE Buku (id_buku int PRIMARY KEY AUTO_INCREMENT, judul_buku
varchar(50), penulis varchar(50), kategori varchar(30), stok int);
```

```
CREATE TABLE Siswa (id_siswa int PRIMARY KEY AUTO_INCREMENT, nama varchar(50), kelas
varchar(50));
```

```
CREATE TABLE Peminjaman (id_peminjaman int PRIMARY KEY AUTO_INCREMENT, id_siswa
int, id_buku int, tanggal_pinjam date, tanggal_kembali date, status varchar(50));
```

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

```
INSERT INTO Buku (judul_buku, penulis, kategori, stok) VALUES
```

```
('Algoritma dan Pemrograman', 'Andi Wijaya', 'Teknologi', 5),
```

```
('Dasar-dasar Database', 'Budi Santoso', 'Teknologi', 7),
```

```
('Matematika Diskrit', 'Rina Sari', 'Matematika', 4),
```

```
('Sejarah Dunia', 'John Smith', 'Sejarah', 3),
```

```
('Pemrograman Web dengan PHP', 'Eko Prasetyo', 'Teknologi', 8);
```

```
INSERT INTO siswa (nama, kelas) VALUES
```

```
('Andi Saputra', 'X-RPL'),
```

```
('Budi Wijaya', 'X-TKJ'),
```

```
('Citra Lestari', 'XI-RPL'),
```

```
('Dewi Kurniawan', 'XI-TKJ'),
```

```
('Eko Prasetyo', 'XII-RPL');
```

```
INSERT INTO peminjaman (id_siswa, id_buku, tanggal_pinjam, tanggal_kembali, status)
VALUES
```

```
(1, 2, '2025-02-01', '2025-02-08', 'Dipinjam'),
```

```
(2, 5, '2025-01-28', '2025-02-04', 'Dikembalikan'),
```

```
(3, 8, '2025-02-02', '2025-02-09', 'Dipinjam'),  
(4, 10, '2025-01-30', '2025-02-06', 'Dikembalikan'),  
(5, 3, '2025-01-25', '2025-02-01', 'Dikembalikan');
```

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

```
DELIMITER //
```

```
CREATE PROCEDURE insertBuku(  
    IN judul_bukuBaru varchar(50),  
    IN penulisBaru varchar(50),  
    IN kategoribaru varchar(50),  
    IN stokBaru int  
)  
BEGIN  
    INSERT INTO buku (judul_buku, penulis, kategori, stok)  
    VALUES (judul_bukuBaru, penulisBaru, kategoriBaru, stokBaru);  
END //
```

```
DELIMITER ;;
```

```
CALL insertBuku('Sistem Operasi', 'Dian Kurniawan', 'Teknologi', 6);
```

```
CALL insertBuku('Jaringan Komputer', 'Ahmad Fauzi', 'Teknologi', 5);
```

```
CALL insertBuku('Cerita Rakyat Nusantara', 'Lestari Dewi', 'Sastra', 9);
```

```
CALL insertBuku('Bahasa Inggris untuk Pemula', 'Jane Doe', 'Bahasa', 10);
```

```
CALL insertBuku('Biologi Dasar', 'Budi Rahman', 'Sains', 7);
```

```
CALL insertBuku('Kimia Organik', 'Siti Aminah', 'Sains', 5);
```

```
CALL insertBuku('Teknik Elektro', 'Ridwan Hakim', 'Teknik', 6);
```

```
CALL insertBuku('Fisika Modern', 'Albert Einstein', 'Sains', 4);
```

```
CALL insertBuku('Manajemen Waktu', 'Steven Covey', 'Pengembangan', 8);
```

```
CALL insertBuku('Strategi Belajar Efektif', 'Tony Buzan', 'Pendidikan', 6);
```

```
DELIMITER //
```

```
CREATE PROCEDURE insertSiswa(  
    IN namaBaru varchar(50),
```

```
        IN kelasBaru varchar(50)
    )
BEGIN
INSERT INTO siswa (nama. kelas)
VALUES (namaBaru, kelasBaru);
END //
```

```
CALL insertSiswa('Farhan Maulana', 'XII-TKJ');
CALL insertSiswa('Gita Permata', 'X-RPL');
CALL insertSiswa('Hadi Sucipto', 'X-TKJ');
CALL insertSiswa('Intan Permadi', 'XI-RPL');
CALL insertSiswa('Joko Santoso', 'XI-TKJ');
CALL insertSiswa('Kartika Sari', 'XII-RPL');
CALL insertSiswa('Lintang Putri', 'XII-TKJ');
CALL insertSiswa('Muhammad Rizky', 'X-RPL');
CALL insertSiswa('Novi Andriana', 'X-TKJ');
CALL insertSiswa('Olivia Hernanda', 'XI-RPL');
```

```
DELIMITER //
```

```
CREATE PROCEDURE insertPeminjaman(
    IN id_siswaBaru int,
    IN id_bukuBaru int,
    IN tanggal_pinjamBaru date,
    IN tanggal_kembaliBaru date,
    IN statusBaru varchar(50)
)
BEGIN
INSERT INTO peminjaman (id_siswa, id_buku, tanggal_pinjam, tanggal_kembali, status)
VALUES (id_siswaBaru, id_bukuBaru, tanggal_pinjamBaru,tanggal_kembaliBaru,statusBaru);
END //
```

```
CALL insertPeminjaman(15, 7, '2025-02-01', '2025-02-08', 'Dipinjam');  
CALL insertPeminjaman(7, 1, '2025-01-29', '2025-02-05', 'Dikembalikan');  
CALL insertPeminjaman(8, 9, '2025-02-03', '2025-02-10', 'Dipinjam');  
CALL insertPeminjaman(13, 4, '2025-01-27', '2025-02-03', 'Dikembalikan');  
CALL insertPeminjaman(10, 11, '2025-02-01', '2025-02-08', 'Dipinjam');
```

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

```
DELIMITER //
```

```
CREATE PROCEDURE updateBuku(  
    IN idBuku INT,  
    IN judulBaru VARCHAR(50),  
    IN penulisBaru VARCHAR(50),  
    IN kategoriBaru VARCHAR(50),  
    IN stokBaru INT  
)
```

```
BEGIN
```

```
    UPDATE buku  
    SET judul_buku = judulBaru,  
        penulis = penulisBaru,  
        kategori = kategoriBaru,  
        stok = stokBaru  
    WHERE id_buku = idBuku;
```

```
END //
```

```
DELIMITER ;
```

```
DELIMITER //
```

```
CREATE PROCEDURE deleteBuku(  
    IN idBuku INT
```

```
)  
BEGIN  
    DELETE FROM buku WHERE id_buku = idBuku;  
END //
```

```
DELIMITER ;
```

```
DELIMITER //
```

```
CREATE PROCEDURE updateSiswa(  
    IN idSiswa INT,  
    IN namaBaru VARCHAR(50),  
    IN kelasBaru VARCHAR(50)  
)
```

```
BEGIN  
    UPDATE siswa  
    SET nama = namaBaru,  
        kelas = kelasBaru  
    WHERE id_siswa = idSiswa;  
END //
```

```
DELIMITER ;
```

```
DELIMITER //
```

```
CREATE PROCEDURE deleteSiswa(  
    IN idSiswa INT  
)  
BEGIN
```

```
DELETE FROM siswa WHERE id_siswa = idSiswa;  
END //
```

```
DELIMITER ;
```

```
DELIMITER //
```

```
CREATE PROCEDURE updatePeminjaman(  
    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(50)  
)  
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//
```

```
CREATE PROCEDURE deletePeminjaman(  
    IN p_id_peminjaman INT  
)  
BEGIN
```



```
DELETE FROM peminjaman WHERE ID_Peminjaman = p_id_peminjaman;  
END //
```

6. Buatlah stored procedure untuk menampilkan seluruh record di setiap table.

```
DELIMITER //
```

```
CREATE PROCEDURE showBuku()
```

```
BEGIN
```

```
SELECT * FROM buku;
```

```
END//
```

```
DELIMITER //
```

```
CREATE PROCEDURE showSiswa()
```

```
BEGIN
```

```
SELECT * FROM siswa;
```

```
END//
```

```
DELIMITER //
```

```
CREATE PROCEDURE showPeminjaman()
```

```
BEGIN
```

```
SELECT * FROM peminjaman;
```

```
END//
```

7. Stok buku pada saat dipinjam berkurang secara otomatis.

```
DELIMITER //
```

```
CREATE TRIGGER after_peminjaman_insert
```

```
AFTER INSERT ON peminjaman
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    UPDATE buku
```

```
    SET stok = stok - 1
```

```
    WHERE id_buku = NEW.id_buku;
```

```
END;
```

```
//
```

8. Stok buku pada saat dikembalikan bertambah secara otomatis.

DELIMITER //

CREATE TRIGGER after_peminjaman_update

AFTER UPDATE ON peminjaman

FOR EACH ROW

BEGIN

-- Jika status berubah menjadi "Dikembalikan", tambahkan stok

IF OLD.status = 'Dipinjam' AND NEW.status = 'Dikembalikan' THEN

UPDATE buku

SET stok = stok + 1

WHERE id_buku = NEW.id_buku;

END IF;

END//

9. Buatlah stored procedure untuk mengembalikan buku dan gunakan tanggal pengembalian sesuai dengan tanggal saat mengembalikan (CURRENT DATE).

DELIMITER //

CREATE PROCEDURE returnBuku(

IN p_id_peminjaman INT

)

BEGIN

UPDATE peminjaman

SET tanggal_kembali = 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 showSiswaPeminjam()

BEGIN

```
SELECT DISTINCT s.id_siswa, s.nama, s.kelas  
FROM siswa s  
JOIN peminjaman p ON s.id_siswa = p.id_siswa;  
END;
```

```
//
```

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,  
           IFNULL(COUNT(p.id_peminjaman), 0) AS jumlah_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;
```

```
//
```

12. Buatlah stored procedure untuk menampilkan semua buku, termasuk yang belum pernah dipinjam.

```
DELIMITER //
```

```
CREATE PROCEDURE showAllBuku()  
BEGIN  
    SELECT b.id_buku, b.judul_buku, b.penulis, b.kategori, b.stok,  
           IFNULL(COUNT(p.id_peminjaman), 0) AS jumlah_dipinjam  
    FROM buku b  
    LEFT JOIN peminjaman p ON b.id_buku = p.id_buku  
    GROUP BY b.id_buku, b.judul_buku, b.penulis, b.kategori, b.stok;
```

END;

//

~ PUSH File SQL ke gitHub dengan nama repository Database-Pepustakaan-Sekolah ~