

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.
CREATE DATABASE db_perpus;
2. Buatlah table buku, siswa dan peminjaman.
CREATE TABLE buku (
 Id_Buku INT AUTO_INCREMENT PRIMARY KEY,
 JudulBuku VARCHAR(50) ,
 Penulis VARCHAR(20) ,
 Kategori VARCHAR(20) ,
 Stok INT
);

CREATE TABLE siswa (
 Id_Siswa INT AUTO_INCREMENT PRIMARY KEY,
 Nama VARCHAR(50) ,
 Kelas VARCHAR(15)
);

CREATE TABLE peminjaman (
 Id_Peminjaman INT AUTO_INCREMENT PRIMARY KEY,
 Id_Siswa INT ,
 Id_Buku INT ,
 TanggalPinjam DATE,
 TanggalKembali DATE,
 Status ENUM('Dipinjam', 'Dikembalikan')
);
3. Input 5 record di setiap table menggunakan query INSERT, UPDATE, DELETE.
INSERT INTO buku (JudulBuku, 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, TanggalPinjam, TanggalKembali, Status)
VALUES
(11, 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');

```

```

UPDATE buku
SET Stok = 6
WHERE Id_Buku = 3;
UPDATE siswa
SET Kelas = 'XII-RPL'
WHERE Id_Siswa=4;
UPDATE peminjaman
SET Id_Siswa=4
WHERE Id_Peminjaman=2;

```

```

DELETE FROM buku WHERE Id_Buku=2;
DELETE FROM siswa WHERE Id_Siswa = 4;
DELETE FROM peminjaman WHERE Id_Peminjaman = 4;

```

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

```

CREATE PROCEDURE InsertBuku(
    IN JudulBukuBaru VARCHAR(50),
    IN PenulisBaru VARCHAR(20),
    IN KategoriBaru VARCHAR(20),
    IN StokBaru INT
)
BEGIN
    INSERT INTO buku(JudulBuku, Penulis, Kategori, Stok)
VALUES(JudulBukuBaru, PenulisBaru, KategoriBaru, StokBaru);
END //
DELIMITER //

```

```
CREATE PROCEDURE InsertSiswa(  
    IN NamaBaru VARCHAR(50),  
    IN KelasBaru VARCHAR(15)  
BEGIN  
    INSERT INTO siswa(Nama,Kelas) VALUES(NamaBaru,KelasBaru);  
END //
```

```
DELIMITER //
```

```
CREATE PROCEDURE InsertPeminjaman(  
    IN Id_SiswaBaru INT,  
    IN Id_BukuBaru INT,  
    IN TanggalPinjamBaru DATE,  
    IN TanggalKembaliBaru DATE,  
    IN StatusBaru ENUM("Dipinjam", "Dikembalikan")  
BEGIN  
    INSERT INTO peminjaman(Id_Siswa,Id_Buku,TanggalPinjam,TanggalKembali,Status)  
VALUES(Id_SiswaBaru,Id_BukuBaru,TanggalPinjamBaru,TanggalKembaliBaru,StatusBaru);  
  
END //
```

```
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);
```

```
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');
```

```
CALL InsertPeminjaman(15, 7, '2025-02-01', 7, '2025-02-08', 'Dipinjam');
```

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

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

```
DELIMITER //
```

```
CREATE PROCEDURE UpdateBuku(
    IN Id_BukuBaru INT,
    IN JudulBukuBaru VARCHAR(50),
    IN PenulisBaru VARCHAR(20),
    IN KategoriBaru VARCHAR(20),
    IN StokBaru INT
)
BEGIN
    UPDATE buku SET JudulBuku=JudulBukuBaru, Penulis=PenulisBaru,
    Kategori=KategoriBaru,Stok=StokBaru WHERE Id_Buku=Id_BukuBaru;
END //
```

```
DELIMITER //
```

```
CREATE PROCEDURE UpdateSiswa(
    IN Id_SiswaBaru INT,
    IN NamaBaru VARCHAR(50),
    IN KelasBaru VARCHAR(15)
)
BEGIN
    UPDATE siswa SET Nama=NamaBaru, Kelas=KelasBaru WHERE Id_Siswa = Id_SiswaBaru;
END //
```

```
DELIMITER //
```

```
CREATE PROCEDURE UpdatePeminjaman(
    IN Id_PeminjamanBaru INT,
    IN Id_SiswaBaru INT,
    IN Id_BukuBaru INT,
    IN TanggalPinjamBaru DATE,
    IN TanggalKembaliBaru DATE,
    IN StatusBaru ENUM("Dipinjam", "Dikembalikan")
)
BEGIN
    UPDATE peminjaman SET Id_Siswa=Id_SiswaBaru, Id_Buku=Id_BukuBaru,
    TanggalPinjam=TanggalPinjamBaru,TanggalKembali=TanggalKembaliBaru, Status=StatusBaru
    WHERE Id_Peminjaman = Id_PeminjamanBaru;
END //
```

```
DELIMITER //
```

```
CREATE PROCEDURE DeleteBuku(  
    IN Id_BukuBaru INT  
)  
BEGIN  
    DELETE FROM buku WHERE Id_Buku = Id_BukuBaru;  
END //
```

```
DELIMITER //
```

```
CREATE PROCEDURE DeleteSiswa(  
    IN Id_SiswaBaru INT  
)  
BEGIN  
    DELETE FROM siswa WHERE Id_Siswa = Id_SiswaBaru;  
END //
```

```
DELIMITER //
```

```
CREATE PROCEDURE UpdatePeminjaman(  
    IN Id_PeminjamanBaru INT  
)  
BEGIN  
    DELETE FROM peminjaman WHERE Id_Peminjaman = Id_PeminjamanBaru;  
END //
```

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

```
DELIMITER //
```

```
CREATE PROCEDURE SelectBuku()  
BEGIN  
    SELECT * FROM buku;  
END //
```

```
DELIMITER //
```

```
CREATE PROCEDURE SelectSiswa()  
BEGIN  
    SELECT * FROM siswa;  
END //
```

```
DELIMITER //
```

```
CREATE PROCEDURE SelectPeminjaman()
```

```
BEGIN
SELECT * FROM peminjaman;
END //
```

7. Stok buku pada saat dipinjam berkurang secara otomatis.

```
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 TambahStokBuku
AFTER UPDATE ON peminjaman
FOR EACH ROW
BEGIN
    IF NEW.status = 'Dikembalikan' AND OLD.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 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 DaftarSiswaPeminjam()
BEGIN
    SELECT DISTINCT s.Id_Siswa, s.Nama, s.Kelas
```

```
FROM siswa s
JOIN peminjaman p ON s.Id_Siswa = p.Id_Siswa
WHERE p.status IN ('Dipinjam', 'Dikembalikan');
END //
```

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

```
DELIMITER //
```

```
CREATE PROCEDURE DaftarSiswaLengkap()
BEGIN
    SELECT s.Id_Siswa, s>Nama, s.Kelas
    FROM siswa s
    LEFT JOIN peminjaman p ON s.Id_Siswa = p.Id_Siswa
    WHERE p.Id_Siswa IS NULL OR p.status IN ('Dipinjam', 'Dikembalikan');
END //
```

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

```
DELIMITER //
```

```
CREATE PROCEDURE DaftarBukuLengkap()
BEGIN
    SELECT b.Id_Buku, b.JudulBuku, b.Penulis, b.Kategori, b.Stok
    FROM buku b
    LEFT JOIN peminjaman p ON b.Id_Buku = p.Id_Buku
    WHERE p.Id_Buku IS NULL OR p.status IN ('Dipinjam', 'Dikembalikan');
END //
```

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