

SISTEM PERPUSTAKAAN SEKOLAH

Rienaldy Davis Manurung XI RPL 2

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

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.

- table buku

```
CREATE TABLE buku ( id_buku INT PRIMARY KEY AUTO_INCREMENT,
judul_buku VARCHAR(255),
penulis VARCHAR(255),
kategori VARCHAR(100),
stok INT );
```

- table siswa

```
CREATE TABLE siswa ( id_siswa INT PRIMARY KEY AUTO_INCREMENT, nama VARCHAR(255), kelas
VARCHAR(50) );
```

- table peminjaman

```
CREATE TABLE peminjaman ( id_peminjaman INT PRIMARY KEY AUTO_INCREMENT, id_siswa INT,
id_buku INT, tanggal_pinjam DATE, tanggal_kembali DATE, status ENUM('Dipinjam', 'Dikembalikan'),
FOREIGN KEY (id_siswa) REFERENCES siswa(id_siswa), FOREIGN KEY (id_buku) REFERENCES
buku(id_buku) );
```

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, 1,
'2025-02-01', '2025-02-08', 'Dipinjam'),
(2, 2, '2025-01-28', '2025-02-04', 'Dikembalikan'),
(3, 3, '2025-02-02', '2025-02-09', 'Dipinjam'),
(4, 4, '2025-01-30', '2025-02-06', 'Dikembalikan'),
(5, 5, '2025-01-25', '2025-02-01', 'Dikembalikan');
```

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

```
DELIMITER //
CREATE PROCEDURE InsertBuku(
    IN p_judul VARCHAR(255),
    IN p_penulis VARCHAR(255),
    IN p_kategori VARCHAR(100),
    IN p_stok INT
)
BEGIN
    INSERT INTO buku (judul_buku, penulis, kategori, stok)
    VALUES (p_judul, p_penulis, p_kategori, p_stok);
END //
DELIMITER ;
```

```
DELIMITER //
CREATE PROCEDURE InsertSiswa(
    IN p_nama VARCHAR(255),
    IN p_kelas VARCHAR(50)
)
BEGIN
    INSERT INTO siswa (nama, kelas)
    VALUES (p_nama, p_kelas);
END //
DELIMITER ;
```

```

DELIMITER //
CREATE PROCEDURE InsertPeminjaman(
    IN p_id_siswa INT,
    IN p_id_buku INT,
    IN p_tanggal_pinjam DATE,
    IN p_tanggal_kembali DATE
)
BEGIN
    INSERT INTO peminjaman (id_siswa, id_buku, tanggal_pinjam, tanggal_kembali, status)
    VALUES (p_id_siswa, p_id_buku, p_tanggal_pinjam, p_tanggal_kembali, 'Dipinjam');

    -- Mengurangi stok buku secara otomatis saat dipinjam
    UPDATE buku SET stok = stok - 1 WHERE id_buku = p_id_buku;
END //
DELIMITER ;

```

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

```

DELIMITER //
CREATE PROCEDURE UpdateBuku
(IN id INT, IN new_stok INT) BEGIN UPDATE buku SET stok = new_stok WHERE id_buku = id;
END //
DELIMITER ;

```

```

DELIMITER //
CREATE PROCEDURE DeleteBuku
(IN id INT) BEGIN DELETE FROM buku WHERE id_buku = id; END //
DELIMITER ;

```

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

- buku

```

DELIMITER //
CREATE PROCEDURE ShowAllBuku()
BEGIN
    SELECT * FROM buku;
END //
DELIMITER ;

```

- siswa

```

DELIMITER //
CREATE PROCEDURE ShowAllSiswa()
BEGIN
    SELECT * FROM siswa;
END //
DELIMITER ;

```

- peminjaman

```
DELIMITER //
CREATE PROCEDURE ShowAllPeminjaman()
BEGIN
    SELECT * FROM peminjaman;
END //
DELIMITER ;
```

7. Stok buku pada saat dipinjam berkurang secara otomatis.

```
DELIMITER //
CREATE TRIGGER kurangi_stok
BEFORE INSERT ON peminjaman
FOR EACH ROW
BEGIN
    UPDATE buku SET stok = stok - 1 WHERE id_buku = NEW.id_buku;
END //
DELIMITER ;
```

8. Stok buku pada saat dikembalikan bertambah secara otomatis.

```
CREATE TRIGGER after_peminjaman_update
AFTER UPDATE ON peminjaman
FOR EACH ROW
BEGIN
    -- Cek jika status diubah menjadi 'Dikembalikan' dari status sebelumnya bukan 'Dikembalikan'
    IF NEW.Status = 'Dikembalikan' AND OLD.Status != 'Dikembalikan' THEN
        -- Tambah stok buku terkait
        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 sp_kembalikan_buku(
    IN p_id_peminjaman INT
)
BEGIN
    -- Update status dan tanggal kembali ke tanggal saat ini
    UPDATE peminjaman
    SET
        Status = 'Dikembalikan',
        Tanggal_Kembali = CURDATE()
    WHERE
        ID_Peminjaman = p_id_peminjaman;
```

END\$\$

DELIMITER ;

10. Buatlah stored procedure untuk menampilkan daftar siswa yang pernah meminjam buku.

```
DELIMITER $$
CREATE PROCEDURE sp_siswa_peminjam()
BEGIN
    SELECT DISTINCT
        siswa.id_siswa,
        siswa.nama,
        siswa.kelas
    FROM
        siswa
    INNER JOIN
        peminjaman ON siswa.id_siswa = peminjaman.id_siswa;
END$$
DELIMITER ;
```

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

```
CREATE PROCEDURE sp_semua_siswa()
BEGIN
    SELECT
        s.*,
        CASE
            WHEN EXISTS (
                SELECT 1
                FROM peminjaman p
                WHERE p.id_siswa = s.id_siswa
            ) THEN 'Pernah Meminjam'
            ELSE 'Tidak Pernah Meminjam'
        END AS Status_Peminjaman
    FROM
        siswa s;
END;
```

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

```
CREATE PROCEDURE sp_semua_buku()
BEGIN
    SELECT
        b.*,
        CASE
            WHEN EXISTS (
                SELECT 1
                FROM peminjaman p
                WHERE p.id_buku = b.id_buku
            ) THEN 'Pernah Dipinjam'
            ELSE 'Belum Pernah Dipinjam'
        END AS Status_Peminjaman
    FROM
        buku b;
END;
```

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