

SISTEM PERPUSTAKAAN SEKOLAH

Muhammad Dhafin Arizki

11 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
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 PRIMARY KEY AUTO_INCREMENT,
  judul_buku VARCHAR(255),
  penulis VARCHAR(255),
  kategori VARCHAR(100),
  stok INT
);
CREATE TABLE siswa (
  id_siswa INT PRIMARY KEY AUTO_INCREMENT,
  nama VARCHAR(255),
  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 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.

```

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
(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');

```

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

```

DELIMITER //
CREATE PROCEDURE InsertDataBuku(
    IN judul_buku VARCHAR(255),
    IN penulis VARCHAR(255),
    IN kategori VARCHAR(100),
    IN stok INT
)
BEGIN
    INSERT INTO buku (judul_buku, penulis, kategori, stok) VALUES (judul_buku, penulis, kategori,
stok);
END //
DELIMITER ;

CALL InsertDataBuku('Sistem Operasi', 'Dian Kurniawan', 'Teknologi', 6);
CALL InsertDataBuku('Jaringan Komputer', 'Ahmad Fauzi', 'Teknologi', 5);
CALL InsertDataBuku('Cerita Rakyat Nusantara', 'Lestari Dewi', 'Sastra', 9);
CALL InsertDataBuku('Bahasa Inggris untuk Pemula', 'Jane Doe', 'Bahasa', 10);
CALL InsertDataBuku('Biologi Dasar', 'Budi Rahman', 'Sains', 7);
CALL InsertDataBuku('Kimia Organik', 'Siti Aminah', 'Sains', 5);
CALL InsertDataBuku('Teknik Elektro', 'Ridwan Hakim', 'Teknik', 6);
CALL InsertDataBuku('Fisika Modern', 'Albert Einstein', 'Sains', 4);
CALL InsertDataBuku('Manajemen Waktu', 'Steven Covey', 'Pengembangan', 8);

```

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

```
DELIMITER //
```

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

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

```
DELIMITER $$
```

```
CREATE PROCEDURE InsertDataPeminjaman(  
    IN id_siswa INT,  
    IN id_buku INT,  
    IN tanggal_pinjam DATE,  
    IN tanggal_kembali DATE,  
    IN status ENUM('Dipinjam', 'Dikembalikan')  
)  
BEGIN  
    INSERT INTO peminjaman (id_siswa, id_buku, tanggal_pinjam, tanggal_kembali, status)  
    VALUES (id_siswa, id_buku, tanggal_pinjam, tanggal_kembali, status);  
END$$  
DELIMITER ;
```

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

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

```
DELIMITER $$
```

```
CREATE PROCEDURE UpdateDataBuku(  
    IN id_buku INT,  
    IN judul_buku VARCHAR(255),
```

```

        IN penulis VARCHAR(255),
        IN kategori VARCHAR(100),
        IN stok INT
    )
BEGIN
    UPDATE buku SET judul_buku = judul_buku, penulis = penulis, kategori = kategori, stok = stok
    WHERE id_buku = id_buku;
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 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 ;

DELIMITER //

CREATE PROCEDURE DeleteBuku(

```

    IN b_id_buku INT

```

)

BEGIN

```

DELETE FROM buku WHERE id_buku = b_id_buku;

```

END //

DELIMITER ;

```

CREATE PROCEDURE DeleteSiswa(

```

```

    IN s_id_siswa INT

```

)

BEGIN

```

DELETE FROM siswa WHERE id_siswa = s_id_siswa;

```

END;

DELIMITER //

CREATE PROCEDURE DeletePeminjaman(

```

    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.

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

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

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

7. Stok buku pada saat dipinjam berkurang secara otomatis.

```
DELIMITER //
CREATE TRIGGER KurangiStok
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.

```
DELIMITER //
CREATE TRIGGER TambahStok
AFTER UPDATE ON peminjaman
FOR EACH ROW
BEGIN
    IF 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 id_peminjaman INT  
)  
BEGIN  
    UPDATE peminjaman  
    SET status = 'Dikembalikan', tanggal_kembali = CURRENT_DATE  
    WHERE id_peminjaman = id_peminjaman;  
END //
```

DELIMITER ;

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

DELIMITER //

```
CREATE PROCEDURE SiswaPernahPinjam()  
BEGIN  
    SELECT DISTINCT siswa.id_siswa, siswa.nama, siswa.kelas  
    FROM siswa  
    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.

DELIMITER //

```
CREATE PROCEDURE SemuaSiswa()  
BEGIN  
    SELECT siswa.id_siswa, siswa.nama, siswa.kelas, IFNULL(peminjaman.id_peminjaman, 'Belum  
    Meminjam') AS Status_Peminjaman  
    FROM siswa  
    LEFT JOIN peminjaman ON siswa.id_siswa = peminjaman.id_siswa;  
END //
```

DELIMITER ;

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

DELIMITER //

```
CREATE PROCEDURE SemuaBuku()  
BEGIN  
    SELECT buku.id_buku, buku.judul_buku, buku.penulis, buku.kategori, buku.stok,  
    IFNULL(peminjaman.id_peminjaman, 'Belum Dipinjam') AS Status_Peminjaman
```

```
FROM buku  
LEFT JOIN peminjaman ON buku.id_buku = peminjaman.id_buku;  
END //  
DELIMITER ;
```

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