

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.

```
MariaDB [(none)]> CREATE DATABASE db_perpus;
```

2. Buatlah table buku, siswa dan peminjaman.

```

MariaDB [db_perpus]> CREATE TABLE buku (
  ->     id_buku INT PRIMARY KEY AUTO_INCREMENT,
  ->     judul_buku VARCHAR(255),
  ->     penulis VARCHAR(255),
  ->     kategori VARCHAR(100),
  ->     stok INT
  -> );
Query OK, 0 rows affected (0.020 sec)

```

```

MariaDB [db_perpus]> CREATE TABLE siswa (
  ->     id_siswa INT PRIMARY KEY AUTO_INCREMENT,
  ->     nama VARCHAR(255),
  ->     kelas VARCHAR(50)
  -> );
Query OK, 0 rows affected (0.041 sec)

```

```

MariaDB [db_perpus]> 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)
  -> );
Query OK, 0 rows affected (0.050 sec)

```

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

```

MariaDB [db_perpus]> 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);
Query OK, 5 rows affected (0.163 sec)
Records: 5  Duplicates: 0  Warnings: 0

```

```

MariaDB [db_perpus]> 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');
Query OK, 5 rows affected (0.011 sec)
Records: 5  Duplicates: 0  Warnings: 0

```

```

MariaDB [db_perpus]> 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');
Query OK, 5 rows affected (0.008 sec)
Records: 5  Duplicates: 0  Warnings: 0

```

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

```
1 DELIMITER //
2 CREATE PROCEDURE InsertDataBuku()
3 BEGIN
4     INSERT INTO buku (judul_buku, penulis, kategori, stok) VALUES
5     ('Sistem Operasi', 'Dian Kurniawan', 'Teknologi', 6),
6     ('Jaringan Komputer', 'Ahmad Fauzi', 'Teknologi', 5),
7     ('Cerita Rakyat Nusantara', 'Lestari Dewi', 'Sastra', 9),
8     ('Bahasa Inggris untuk Pemula', 'Jane Doe', 'Bahasa', 10),
9     ('Biologi Dasar', 'Budi Rahman', 'Sains', 7),
10    ('Kimia Organik', 'Siti Aminah', 'Sains', 5),
11    ('Teknik Elektro', 'Ridwan Hakim', 'Teknik', 6),
12    ('Fisika Modern', 'Albert Einstein', 'Sains', 4),
13    ('Manajemen Waktu', 'Steven Covey', 'Pengembangan', 8),
14    ('Strategi Belajar Efektif', 'Tony Buzan', 'Pendidikan', 6);
15 END //
16 DELIMITER ;
17
18 CALL InsertDataBuku();
```

```
1 DELIMITER //
2 CREATE PROCEDURE InsertDataSiswa()
3 BEGIN
4     INSERT INTO siswa (nama, kelas) VALUES
5     ('Ahmad Fauzan', 'X-RPL'),
6     ('Budi Santoso', 'X-TKJ'),
7     ('Citra Permata', 'XI-RPL'),
8     ('Dewi Kartika', 'XI-TKJ'),
9     ('Eko Saputra', 'XII-RPL'),
10    ('Farhan Malik', 'XII-TKJ'),
11    ('Gita Sari', 'X-RPL'),
12    ('Hendra Wijaya', 'X-TKJ'),
13    ('Indah Lestari', 'XI-RPL'),
14    ('Joko Prasetyo', 'XI-TKJ');
15 END //
16 DELIMITER ;
```

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

```
1 DELIMITER //
2 CREATE PROCEDURE update_data_buku(IN id INT, IN new_stok INT)
3 BEGIN
4     UPDATE buku SET stok = n_stok WHERE id_buku = id;
5 END //
6 DELIMITER ;
```

```
1 DELIMITER //
```

```
2 CREATE PROCEDURE delete_data_buku(IN id INT)
```

```
3 BEGIN
```

```
4     DELETE FROM buku WHERE id_buku = id;
```

```
5 END //
```

```
6 DELIMITER ;
```

```
1 DELIMITER //
```

```
2 CREATE PROCEDURE UpdateSiswa(
```

```
3     IN id INT,
```

```
4     IN new_nama VARCHAR(255),
```

```
5     IN new_kelas VARCHAR(50)
```

```
6 )
```

```
7 BEGIN
```

```
8     UPDATE siswa
```

```
9     SET nama = new_nama, kelas = new_kelas
```

```
10    WHERE id_siswa = id;
```

```
11 END //
```

```
12 DELIMITER ;
```

```
1 DELIMITER //
```

```
2 CREATE PROCEDURE DeleteSiswa(IN id INT)
```

```
3 BEGIN
```

```
4     DELETE FROM siswa WHERE id_siswa = id;
```

```
5 END //
```

```
6 DELIMITER ;
```

```
1 DELIMITER //
```

```
2 CREATE PROCEDURE UpdatePeminjaman(
```

```
3     IN id INT,
```

```
4     IN new_status ENUM('Dipinjam', 'Dikembalikan')
```

```
5 )
```

```
6 BEGIN
```

```
7     UPDATE peminjaman
```

```
8     SET status = new_status
```

```
9     WHERE id_peminjaman = id;
```

```
10 END //
```

```
11 DELIMITER ;
```

```
1 DELIMITER //
```

```
2 CREATE PROCEDURE DeletePeminjaman(IN id INT)
```

```
3 BEGIN
```

```
4     DELETE FROM peminjaman WHERE id_peminjaman = id;
```

```
5 END //
```

```
6 DELIMITER ;
```

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

```
1 DELIMITER //
2 CREATE PROCEDURE ShowAllBuku()
3 BEGIN
4     SELECT * FROM buku;
5 END //
6 DELIMITER ;
```

```
1 DELIMITER //
2 CREATE PROCEDURE ShowAllSiswa()
3 BEGIN
4     SELECT * FROM siswa;
5 END //
6 DELIMITER ;
```

```
1 DELIMITER //
2 CREATE PROCEDURE ShowAllPeminjaman()
3 BEGIN
4     SELECT * FROM peminjaman;
5 END //
6 DELIMITER ;
```

7. Stok buku pada saat dipinjam berkurang secara otomatis.

```
1 DELIMITER //
2 CREATE TRIGGER kurangi_stok
3 BEFORE INSERT ON peminjaman
4 FOR EACH ROW
5 BEGIN
6     UPDATE buku SET stok = stok - 1 WHERE id_buku = NEW.id_buku;
7 END //
8 DELIMITER ;
```

8. Stok buku pada saat dikembalikan bertambah secara otomatis.

```
1 DELIMITER //
2 CREATE TRIGGER tambah_stok
3 BEFORE UPDATE ON peminjaman
4 FOR EACH ROW
5 BEGIN
6     IF NEW.status = 'Dikembalikan' AND OLD.status = 'Dipinjam' THEN
7         UPDATE buku SET stok = stok + 1 WHERE id_buku = NEW.id_buku;
8     END IF;
9 END //
10 DELIMITER ;
```

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

```
1 DELIMITER //
2 CREATE PROCEDURE KembalikanBuku(IN id_pinjam INT)
3 BEGIN
4     UPDATE peminjaman
5     SET status = 'Dikembalikan', tanggal_kembali = CURDATE()
6     WHERE id_peminjaman = id_pinjam;
7 END //
8 DELIMITER ;
```

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

```
1 DELIMITER //
2 CREATE PROCEDURE SiswaPernahPinjam()
3 BEGIN
4     SELECT DISTINCT s.id_siswa, s.nama, s.kelas
5     FROM siswa s
6     JOIN peminjaman p ON s.id_siswa = p.id_siswa;
7 END //
8 DELIMITER ;
```

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

```
1 DELIMITER //
2 CREATE PROCEDURE SemuaSiswa()
3 BEGIN
4     SELECT s.id_siswa, s.nama, s.kelas,
5     IF(p.id_siswa IS NULL, 'Belum Pernah Meminjam', 'Pernah Meminjam') AS Status
6     FROM siswa s
7     LEFT JOIN peminjaman p ON s.id_siswa = p.id_siswa;
8 END //
9 DELIMITER ;
```

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

```
1 DELIMITER //
2 CREATE PROCEDURE SemuaBuku()
3 BEGIN
4     SELECT b.id_buku, b.judul_buku, b.penulis, b.kategori, b.stok,
5     IF(p.id_buku IS NULL, 'Belum Pernah Dipinjam', 'Pernah Dipinjam') AS Status
6     FROM buku b
7     LEFT JOIN peminjaman p ON b.id_buku = p.id_buku;
8 END //
9 DELIMITER ;
```