

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.
2. Buatlah table buku, siswa dan peminjaman.
3. Input 5 record di setiap table menggunakan query INSERT, UPDATE, DELETE.
4. Input 10 record di setiap table menggunakan stored procedure INSERT.
5. Buatlah stored procedure UPDATE, DELETE di setiap table.
6. Buatlah stored procedure untuk menampilkan seluruh record di setiap table.
7. Stok buku pada saat dipinjam berkurang secara otomatis.
8. Stok buku pada saat dikembalikan bertambah secara otomatis.
9. Buatlah stored procedure untuk mengembalikan buku dan gunakan tanggal pengembalian sesuai dengan tanggal saat mengembalikan (CURRENT DATE).
10. Buatlah stored procedure untuk menampilkan daftar siswa yang pernah meminjam buku.
11. Buatlah stored procedure untuk menampilkan semua siswa, termasuk yang tidak pernah meminjam buku.
12. Buatlah stored procedure untuk menampilkan semua buku, termasuk yang belum pernah dipinjam.

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

JAWABAN

1.

```
mysql> CREATE DATABASE db_perpus;  
Query OK, 1 row affected (0.01 sec)
```

2.

```
Database changed  
mysql> CREATE TABLE buku (id_buku INT PRIMARY KEY, judul_buku VARCHAR(255), penulis VARCHAR(100), kategori VARCHAR(50), stok INT);  
Query OK, 0 rows affected (0.06 sec)
```

```
mysql> CREATE TABLE siswa (id_siswa INT PRIMARY KEY, nama VARCHAR(100), kelas VARCHAR(50));  
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> CREATE TABLE peminjaman (id_peminjaman INT PRIMARY KEY, 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 (0.13 sec)
```

3

```
mysql> INSERT INTO buku (id_buku, judul_buku, penulis, kategori, stok) VALUES (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);
```

```
mysql> INSERT INTO siswa (id_siswa, nama, kelas) VALUES (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');  
Query OK, 5 rows affected (0.01 sec)  
Records: 5 Duplicates: 0 Warnings: 0
```

```
mysql> INSERT INTO peminjaman (id_peminjaman, id_siswa, id_buku, tanggal_pinjam, tanggal_kembali, status) VALUES (1, 1, 2, '2025-02-01', '2025-02-08', 'Dipinjam'), (2, 2, 5, '2025-01-28', '2025-02-04', 'Dikembalikan'), (3, 3, 3, '2025-02-02', '2025-02-09', 'Dipinjam'), (4, 4, 4, '2025-01-30', '2025-02-06', 'Dikembalikan'), (5, 5, 3, '2025-01-25', '2025-02-01', 'Dikembalikan');
```

4.

```
DELIMITER //
```

```
CREATE PROCEDURE InsertBuku(  
    IN p_id_buku INT,  
    IN p_judul_buku VARCHAR(255),  
    IN p_penulis VARCHAR(100),  
    IN p_kategori VARCHAR(50),  
    IN p_stok INT  
)  
BEGIN  
    INSERT INTO buku (id_buku, judul_buku, penulis, kategori, stok)  
    VALUES (p_id_buku, p_judul_buku, p_penulis, p_kategori, p_stok);  
END //
```

```
1 CALL InsertBuku(6, 'Sistem Operasi', 'Dian Kurniawan', 'Teknologi', 6);
2 CALL InsertBuku(7, 'Jaringan Komputer', 'Ahmad Fauzi', 'Teknologi', 5);
3 CALL InsertBuku(8, 'Cerita Rakyat Nusantara', 'Lestari Dewi', 'Sastra', 9);
4 CALL InsertBuku(9, 'Bahasa Inggris untuk Pemula', 'Jane Doe', 'Bahasa', 10);
5 CALL InsertBuku(10, 'Biologi Dasar', 'Budi Rahman', 'Sains', 7);
6 CALL InsertBuku(11, 'Kimia Organik', 'Siti Aminah', 'Sains', 5);
7 CALL InsertBuku(12, 'Teknik Elektro', 'Ridwan Hakim', 'Teknik', 6);
8 CALL InsertBuku(13, 'Fisika Modern', 'Albert Einstein', 'Sains', 4);
9 CALL InsertBuku(14, 'Manajemen Waktu', 'Steven Covey', 'Pengembangan', 8);
10 CALL InsertBuku(15, 'Strategi Belajar Efektif', 'Tony Buzan', 'Pendidikan', 6);
1
```

DELIMITER //

```
CREATE PROCEDURE InsertSiswa(
    IN p_id_siswa INT,
    IN p_nama VARCHAR(100),
    IN p_kelas VARCHAR(50)
)
BEGIN
    INSERT INTO siswa (id_siswa, nama, kelas)
    VALUES (p_id_siswa, p_nama, p_kelas);
END //
```

DELIMITER ;

```
CALL InsertSiswa(6, 'Farhan Maulana', 'XII-TKJ');
CALL InsertSiswa(7, 'Gita Permata', 'X-RPL');
CALL InsertSiswa(8, 'Hadi Sucipto', 'X-TKJ');
CALL InsertSiswa(9, 'Intan Permadi', 'XI-RPL');
CALL InsertSiswa(10, 'Joko Santoso', 'XI-TKJ');
CALL InsertSiswa(11, 'Kartika Sari', 'XII-RPL');
CALL InsertSiswa(12, 'Lintang Putri', 'XII-TKJ');
CALL InsertSiswa(13, 'Muhammad Rizky', 'X-RPL');
CALL InsertSiswa(14, 'Novi Andriana', 'X-TKJ');
CALL InsertSiswa(15, 'Olivia Hernanda', 'XI-RPL');
```

```

DELIMITER //

CREATE PROCEDURE InsertPeminjaman(
    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
    INSERT INTO peminjaman (id_peminjaman, id_siswa, id_buku, tanggal_pinjam,
tanggal_kembali, status)
    VALUES (p_id_peminjaman, p_id_siswa, p_id_buku, p_tanggal_pinjam, p_tanggal_kembali,
p_status);
END //

DELIMITER ;

CALL InsertPeminjaman(6, 15, 7, '2025-02-01', '2025-02-08', 'Dipinjam');
CALL InsertPeminjaman(7, 7, 1, '2025-01-29', '2025-02-05', 'Dikembalikan');
CALL InsertPeminjaman(8, 8, 9, '2025-02-03', '2025-02-10', 'Dipinjam');
CALL InsertPeminjaman(9, 13, 4, '2025-01-27', '2025-02-03', 'Dikembalikan');
CALL InsertPeminjaman(10, 10, 11, '2025-02-01', '2025-02-08', 'Dipinjam');

```

5

```

1 DELIMITER //
2
3 CREATE PROCEDURE UpdateBuku(
4     IN p_id_buku INT,
5     IN p_judul_buku VARCHAR(255),
6     IN p_penulis VARCHAR(100),
7     IN p_kategori VARCHAR(50),
8     IN p_stok INT
9 )
10 BEGIN
11     UPDATE buku
12     SET judul_buku = p_judul_buku, penulis = p_penulis, kategori = p_kategori, stok =
p_stok
13     WHERE id_buku = p_id_buku;
14 END //
15
16 DELIMITER ;
17

```

```

1 DELIMITER //
2
3 CREATE PROCEDURE UpdateSiswa(
4     IN p_id_siswa INT,
5     IN p_nama VARCHAR(100),
6     IN p_kelas VARCHAR(50)
7 )
8 BEGIN
9     UPDATE siswa
10    SET nama = p_nama, kelas = p_kelas
11    WHERE id_siswa = p_id_siswa;
12 END //
13
14 DELIMITER ;
15

```

```

1 DELIMITER //
2
3 CREATE PROCEDURE UpdatePeminjaman(
4     IN p_id_peminjaman INT,
5     IN p_id_siswa INT,
6     IN p_id_buku INT,
7     IN p_tanggal_pinjam DATE,
8     IN p_tanggal_kembali DATE,
9     IN p_status ENUM('Dipinjam', 'Dikembalikan')
10 )
11 BEGIN
12     UPDATE peminjaman
13    SET id_siswa = p_id_siswa, id_buku = p_id_buku, tanggal_pinjam = p_tanggal_pinjam,
14    tanggal_kembali = p_tanggal_kembali, status = p_status
15    WHERE id_peminjaman = p_id_peminjaman;
16 END //
17 DELIMITER ;

```

```

1 DELIMITER //
2
3 CREATE PROCEDURE DeleteBuku(
4     IN p_id_buku INT
5 )
6 BEGIN
7     DELETE FROM buku WHERE id_buku = p_id_buku;
8 END //
9
10 DELIMITER ;
11

```

```

1 CREATE PROCEDURE `DeletePeminjaman`(IN `p_id_peminjaman` INT) NOT DETERMINISTIC CONTAINS SQL SQL SECURITY DEFINER
2 BEGIN
3     DELETE
4 FROM
5     peminjaman
6 WHERE
7     id_peminjaman = p_id_peminjaman;
8 END

```

```

1 DROP
2 PROCEDURE `DeleteSiswa`;
3 CREATE DEFINER = `root`@`localhost` PROCEDURE `DeleteSiswa`(IN `p_id_siswa` INT) NOT DETERMINISTIC CONTAINS SQL SQL SECURITY DEFINER
4 BEGIN
5     DELETE
6 FROM
7     siswa
8 WHERE
9     id_siswa = p_id_siswa;
0 END

```

6.

```

1 DELIMITER //
2
3 CREATE PROCEDURE DisplayTable(
4     IN nama_table VARCHAR(100)
5 )
6 BEGIN
7     SELECT * FROM nama_table;
8 END //
9

```

7.

```

DELIMITER //

CREATE TRIGGER stok_berkurang AFTER UPDATE
ON
    peminjaman FOR EACH ROW
BEGIN
    IF OLD.STATUS = 'Dikembalikan' AND NEW.STATUS = 'Dipinjam' THEN
        UPDATE buku
        SET stok = stok - 1
        WHERE id_buku = NEW.id_buku ;
    END IF ;
END //

DELIMITER ;

```

8.

```
1 DELIMITER //
```

```
2
```

```
3 CREATE TRIGGER stok_bertambah
```

```
4 AFTER UPDATE ON peminjaman
```

```
5 FOR EACH ROW
```

```
6 BEGIN
```

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

```
8         UPDATE buku
```

```
9         SET stok = stok + 1
```

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

```
1    END IF;
```

```
2 END //
```

```
3
```

```
4 DELIMITER ;
```

9.

```
DELIMITER //
```

```
CREATE PROCEDURE KembalikanBuku(
```

```
    IN p_id_peminjaman INT
```

```
)
```

```
BEGIN
```

```
    UPDATE peminjaman
```

```
    SET Tanggal_Kembali = CURDATE(),
```

```
        Status = 'Dikembalikan'
```

```
    WHERE ID_Peminjaman = p_id_peminjaman;
```

```
END //
```

```
DELIMITER ;
```

10.

```
DELIMITER //
```

```
CREATE PROCEDURE TampilkanSemuaSiswa()
```

```
BEGIN
```

```
    SELECT s.ID_Siswa, s>Nama, IFNULL(COUNT(p.ID_Peminjaman), 0) AS TotalPeminjaman
```

```
    FROM siswa s
```

```
    LEFT JOIN peminjaman p ON s.ID_Siswa = p.ID_Siswa
```

```
    GROUP BY s.ID_Siswa;
```

```
END //
```

```
DELIMITER ;
```


11.

```
1 DELIMITER //  
2  
3 CREATE PROCEDURE DisplayTable(  
4     IN nama_table VARCHAR(64)  
5 )  
6 BEGIN  
7     SELECT * FROM nama_table;  
8 END //  
9 DELIMITER ;  
10
```

Clear Format Get auto-saved query

12.

```
1 DELIMITER //  
2  
3 CREATE PROCEDURE DisplayTable(  
4     IN nama_table VARCHAR(64)  
5 )  
6 BEGIN  
7     SELECT * FROM nama_table;  
8 END //  
9 DELIMITER ;  
10
```

Clear Format Get auto-saved query