

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.

```
Run SQL query/queries on server "localhost":  
  
1 CREATE DATABASE db_perpus;
```

2. Buatlah table buku, siswa dan peminjaman.

-Tabel Buku

```
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0118 seconds.)  
  
CREATE TABLE buku ( id_buku INT AUTO_INCREMENT PRIMARY KEY, judul_buku VARCHAR(100), penulis VARCHAR(100), kategori VARCHAR(50), stok INT );  
[ Edit inline ] [ Edit ] [ Create PHP code ]
```

-Tabel Siswa

```
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0099 seconds.)  
  
CREATE TABLE siswa ( id_siswa INT AUTO_INCREMENT PRIMARY KEY, nama VARCHAR(100), kelas VARCHAR(50) );  
[ Edit inline ] [ Edit ] [ Create PHP code ]
```

-Tabel Peminjaman

```
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0090 seconds.)  
  
CREATE TABLE peminjaman ( id_peminjaman INT AUTO_INCREMENT PRIMARY KEY, id_siswa INT, id_buku INT, tanggal_pinjam DATE, tanggal_kembali DATE, status ENUM('Dipinjam', 'Dikembalikan') );  
[ Edit inline ] [ Edit ] [ Create PHP code ]
```

3. Input 5 record di setiap table menggunakan query INSERT.

- Tabel Buku

```
5 rows inserted.  
Inserted row id: 5 (Query took 0.0023 seconds.)  
  
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);  
[ Edit inline ] [ Edit ] [ Create PHP code ]
```

-Tabel Siswa

✓ 5 rows inserted.
Inserted row id: 5 (Query took 0.0025 seconds.)

```
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');
```

[Edit inline] [Edit] [Create PHP code]

-Tabel Peminjaman

✓ 5 rows inserted.
Inserted row id: 5 (Query took 0.0016 seconds.)

```
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');
```

[Edit inline] [Edit] [Create PHP code]

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

-Tabel Buku

Procedure Insert Buku :

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0036 seconds.)

```
CREATE PROCEDURE insertBuku( IN judul_buku VARCHAR(100), IN penulis VARCHAR(100), IN kategori VARCHAR(50), IN stok INT ) BEGIN INSERT INTO buku (judul_buku, penulis, kategori, stok) VALUES (judul_buku, penulis, kategori, stok); END;
```

[Edit inline] [Edit] [Create PHP code]

Memasukan Record :

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

-Tabel Siswa

Procedure Insert Siswa :

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0049 seconds.)

```
CREATE PROCEDURE insertSiswa( IN nama VARCHAR(100), IN kelas VARCHAR(100) ) BEGIN INSERT INTO siswa (nama, kelas) VALUES (nama, kelas); END;
```

[Edit inline] [Edit] [Create PHP code]

Memasukan Record :

```
1 CALL insertSiswa('Farhan Maulana', 'XII-TKJ');
2 CALL insertSiswa('Gita Permata', 'X-RPL');
3 CALL insertSiswa('Hadi Sucipto', 'X-TKJ');
4 CALL insertSiswa('Intan Permadi', 'XI-RPL');
5 CALL insertSiswa('Joko Santoso', 'XI-TKJ');
6 CALL insertSiswa('Kartika Sari', 'XII-RPL');
7 CALL insertSiswa('Lintang Putri', 'XII-TKJ');
8 CALL insertSiswa('Muhammad Rizky', 'X-RPL');
9 CALL insertSiswa('Novi Andriana', 'X-TKJ');
10 CALL insertSiswa('Olivia Hernanda', 'XI-RPL');
```

-Tabel Peminjaman

Procedure Insert Peminjaman :

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0019 seconds.)

```
CREATE PROCEDURE insert_peminjaman( 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_siswa, id_buku, tanggal_pinjam, tanggal_kembali, status) VALUES (p_id_siswa, p_id_buku, p_tanggal_pinjam, p_tanggal_kembali, p_status); END;
```

[Edit inline] [Edit] [Create PHP code]

Memasukan Record :

```
1 CALL insert_peminjaman(6, 6, '2025-02-01', '2025-02-08', 'Dipinjam');  
2 CALL insert_peminjaman(7, 7, '2025-02-02', '2025-02-09', 'Dipinjam');  
3 CALL insert_peminjaman(8, 8, '2025-01-30', '2025-02-06', 'Dipinjam');  
4 CALL insert_peminjaman(9, 9, '2025-01-29', '2025-02-05', 'Dipinjam');  
5 CALL insert_peminjaman(10, 10, '2025-02-03', '2025-02-10', 'Dipinjam');
```

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

-Tabel Buku

Procedure UPDATE

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0042 seconds.)

```
CREATE PROCEDURE UpdateBuku( IN book_id INT, IN new_judul VARCHAR(100), IN new_penulis VARCHAR(100), IN new_kategori VARCHAR(50), IN new_stok INT ) BEGIN UPDATE buku SET  
judul_buku = new_judul, penulis = new_penulis, kategori = new_kategori, stok = new_stok WHERE id_buku = book_id; END;
```

[Edit inline] [Edit] [Create PHP code]

Procedure DELETE

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0046 seconds.)

```
CREATE PROCEDURE DeleteBuku(IN book_id INT) BEGIN DELETE FROM buku WHERE id_buku = book_id; END;
```

[Edit inline] [Edit] [Create PHP code]

-TABEL SISWA

Procedure UPDATE

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0047 seconds.)

```
CREATE PROCEDURE UpdateSiswa( IN siswa_id INT, IN new_nama VARCHAR(100), IN new_kelas VARCHAR(50) ) BEGIN UPDATE siswa SET nama = new_nama, kelas = new_kelas WHERE id_siswa =  
siswa_id; END;
```

[Edit inline] [Edit] [Create PHP code]

Procedure DELETE

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0042 seconds.)

```
CREATE PROCEDURE DeleteSiswa(IN siswa_id INT) BEGIN DELETE FROM siswa WHERE id_siswa = siswa_id; END;
```

[Edit inline] [Edit] [Create PHP code]

-TABEL PEMINJAMAN

Procedure UPDATE

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0066 seconds.)

```
CREATE PROCEDURE UpdatePeminjaman( IN pinjam_id INT, IN new_siswa_id INT, IN new_buku_id INT, IN new_tanggal_pinjam DATE, IN new_tanggal_kembali DATE, IN new_status  
ENUM('Dipinjam', 'Dikembalikan') ) BEGIN UPDATE peminjaman SET id_siswa = new_siswa_id, id_buku = new_buku_id, tanggal_pinjam = new_tanggal_pinjam, tanggal_kembali =  
new_tanggal_kembali, status = new_status WHERE id_peminjaman = pinjam_id; END;
```

[Edit inline] [Edit] [Create PHP code]

Procedure DELETE

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0037 seconds.)

```
CREATE PROCEDURE DeletePeminjaman(IN pinjam_id INT) BEGIN DELETE FROM peminjaman WHERE id_peminjaman = pinjam_id; END;
```

[Edit inline] [Edit] [Create PHP code]

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

-TABEL BUKU

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0026 seconds.)

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

[Edit inline] [Edit] [Create PHP code]

-TABEL SISWA

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0050 seconds.)

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

[Edit inline] [Edit] [Create PHP code]

-TABEL PEMINJAMAN

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0043 seconds.)

```
CREATE PROCEDURE GetAllPeminjaman() BEGIN SELECT * FROM peminjaman; END;
```

[Edit inline] [Edit] [Create PHP code]

7. Stok buku pada saat dipinjam berkurang secara otomatis.

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0062 seconds.)

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

[Edit inline] [Edit] [Create PHP code]

8. Stok buku pada saat dikembalikan bertambah secara otomatis.

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0078 seconds.)

```
CREATE TRIGGER TambahStokBuku BEFORE UPDATE ON peminjaman FOR EACH ROW BEGIN -- Menambah stok hanya jika status berubah dari 'Dipinjam' ke 'Dikembalikan' IF OLD.status = 'Dipinjam' AND NEW.status = 'Dikembalikan' THEN UPDATE buku SET stok = stok + 1 WHERE id_buku = NEW.id_buku; END IF; END;
```

[Edit inline] [Edit] [Create PHP code]

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

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0051 seconds.)

```
CREATE PROCEDURE KembalikanBuku(IN pinjam_id INT) BEGIN UPDATE peminjaman SET tanggal_kembali = CURRENT_DATE, status = 'Dikembalikan' WHERE id_peminjaman = pinjam_id; END;
```

[Edit inline] [Edit] [Create PHP code]

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

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0136 seconds.)

```
CREATE PROCEDURE SiswaPeminjam() BEGIN SELECT DISTINCT s.id_siswa, s.nama, s.kelas FROM siswa s JOIN peminjaman p ON s.id_siswa = p.id_siswa; END;
```

[Edit inline] [Edit] [Create PHP code]

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

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0049 seconds.)

```
CREATE PROCEDURE SiswaTermasukNonPeminjam() BEGIN SELECT s.id_siswa, s.nama, s.kelas, IFNULL(p.id_peminjaman, 'Belum Pernah Meminjam') AS status_peminjaman FROM siswa s LEFT JOIN peminjaman p ON s.id_siswa = p.id_siswa; END;
```

[Edit inline] [Edit] [Create PHP code]

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

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0039 seconds.)

```
CREATE PROCEDURE BukuTermasukNonDipinjam() BEGIN SELECT b.id_buku, b.judul_buku, b.penulis, b.kategori, b.stok, IFNULL(p.id_peminjaman, 'Belum Pernah Dipinjam') AS status_peminjaman FROM buku b LEFT JOIN peminjaman p ON b.id_buku = p.id_buku; END;
```

[Edit inline] [Edit] [Create PHP code]