

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

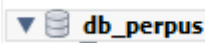
3. Tabel Peminjaman

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.

```
create database db_perpus;
```



2. Buatlah table buku, siswa dan peminjaman.

Tabel Buku

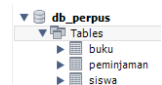
```
CREATE TABLE buku (
  id_buku int primary key auto_increment, judul_buku varchar(50), penulis varchar(50),
  kategori varchar(50), stok int);
```

Tabel Siswa

```
CREATE TABLE siswa (
  id_siswa int primary key auto_increment, nama varchar(50), kelas varchar(50));
```

Tabel Peminjaman

```
CREATE TABLE peminjaman(
  id_peminjaman int primary key auto_increment, id_siswa int, id_buku int,
  tgl_pinjam date, tgl_kembali date, status varchar(20));
```



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

Tabel Buku

```
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");
```

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

Tabel Siswa

```
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");
```

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
NULL	NULL	NULL

Tabel Peminjaman

```
INSERT INTO peminjaman (id_siswa,id_buku,tgl_pinjam,tgl_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');
```

id_peminjaman	id_siswa	id_buku	tgl_pinjam	tgl_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
NULL	NULL	NULL	NULL	NULL	NULL

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

Tabel Buku

```
DELIMITER $$
CREATE PROCEDURE insert_buku(
pJudul_buku varchar(50), pPenulis varchar(50), pKategori varchar(50), pStok int)
BEGIN
INSERT INTO buku (judul_buku, penulis, kategori, stok) VALUES (pJudul_buku, pPenulis,pKategori, pStok);
END $$
DELIMITER ;
```

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

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	Pengemba...	8
15	Strategi Belajar Efektif	Tony Buzan	Pendidikan	6

Tabel Siswa

```
DELIMITER $$
CREATE PROCEDURE insert_siswa(
pNama varchar(50), pKelas varchar(50))
BEGIN
INSERT INTO siswa (nama,kelas) VALUES (pNama,pKelas);
END $$
DELIMITER ;

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

Tabel Peminjaman

```
DELIMITER $$
CREATE PROCEDURE insert_peminjaman(
  pId_siswa int, pId_buku int, pTgl_pinjam date, pTgl_kembali date, pStatus varchar(20))
BEGIN
  INSERT INTO peminjaman (id_siswa, id_buku, tgl_pinjam, tgl_kembali, status) VALUES (pId_siswa, pId_buku, pTgl_pinjam, pTgl_kembali, pStatus);
END $$
DELIMITER ;

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

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

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

Tabel Buku

UPDATE

```
DELIMITER $$
CREATE PROCEDURE update_buku(
  pId_buku int, pJudul varchar(50), pPenulis varchar(50), pKategori varchar(50), pStok int)
BEGIN
  UPDATE buku SET judul = pJudul, penulis = pPenulis, kategori = pKategori, stok = pStok WHERE id_buku = pId_buku;
END $$
DELIMITER ;
```

DELETE

```
DELIMITER $$
CREATE PROCEDURE update_buku(
  pId_buku int, pJudul varchar(50), pPenulis varchar(50), pKategori varchar(50), pStok int)
BEGIN
  UPDATE buku SET judul = pJudul, penulis = pPenulis, kategori = pKategori, stok = pStok WHERE id_buku = pId_buku;
END $$
DELIMITER ;
```

Tabel Siswa

UPDATE

```
DELIMITER $$
CREATE PROCEDURE update_siswa(
  pId_siswa int, pNama varchar(50), pKelas varchar(50))
BEGIN
  UPDATE siswa SET nama = pNama, kelas = pKelas WHERE id_siswa = pId_siswa;
END $$
DELIMITER ;
```

DELETE

```
DELIMITER $$
CREATE PROCEDURE delete_siswa(
  pId_siswa int)
BEGIN
  DELETE FROM siswa WHERE id_siswa = pId_siswa;
END $$
DELIMITER ;
```

Tabel Peminjaman

UPDATE

```
DELIMITER $$
CREATE PROCEDURE update_peminjaman(
  pId_peminjaman int, pId_siswa int, pId_buku int, pTgl_pinjam date, pTgl_kembali date, pStatus date)
BEGIN
  UPDATE peminjaman SET id_siswa = pId_siswa, kelas = pId_buku, tgl_pinjam = pTgl_pinjam, tgl_kembali = pTgl_kembali, status = pStatus WHERE id_peminjaman = pId_peminjaman;
END $$
DELIMITER ;
```

DELETE

```
DELIMITER $$
CREATE PROCEDURE delete_peminjaman(
  pId_peminjaman int)
BEGIN
  DELETE FROM peminjaman WHERE id_peminjaman = pId_peminjaman;
END $$
DELIMITER ;
```

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

```
DELIMITER $$
CREATE PROCEDURE tampil_seluruh()
BEGIN
    SELECT * FROM buku;
    SELECT * FROM siswa;
    SELECT * FROM peminjaman;
END $$
DELIMITER ;
```

7. Stok buku pada saat dipinjam berkurang secara otomatis.

```
DELIMITER $$
CREATE TRIGGER after_update_buku AFTER UPDATE ON buku
FOR EACH ROW
BEGIN
    IF NEW.status = 'Dipinjam' AND OLD.status <> 'Dipinjam' THEN
        UPDATE buku SET stok = stok - 1 WHERE id_buku = NEW.id_buku;
    END IF;
END $$
DELIMITER ;
```

8. Stok buku pada saat dikembalikan bertambah secara otomatis.

```
DELIMITER $$
CREATE TRIGGER after_update_buku_pinjam AFTER UPDATE ON buku
FOR EACH ROW
BEGIN
    IF NEW.status = 'Dikembalikan' AND OLD.status <> 'Dikembalikan' THEN
        UPDATE buku SET stok = stok + 1 WHERE id_buku = NEW.id_buku;
    END IF;
END $$
DELIMITER ;
```

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

```
DELIMITER $$
CREATE PROCEDURE kembali_buku(
    pId_peminjaman int, pId_buku int)
BEGIN
    UPDATE peminjaman SET status = 'Dikembalikan', tgl_kembali = CURRENT_DATE WHERE id_peminjaman = pId_peminjaman;
END $$
DELIMITER ;
```

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

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

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

```
DELIMITER $$
CREATE PROCEDURE tampil_semua_siswa()
BEGIN
    SELECT siswa.id_siswa, siswa.nama, siswa.kelas,
        IFNULL(COUNT(peminjaman.id_peminjaman), 0) AS jumlah_peminjaman
    FROM siswa LEFT JOIN peminjaman ON siswa.id_siswa = peminjaman.id_siswa GROUP BY siswa.id_siswa, siswa.nama, siswa.kelas;
END $$
DELIMITER ;
```

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

```
DELIMITER $$
CREATE PROCEDURE tampil_semua_buku()
BEGIN
    SELECT buku.id_buku, buku.judul_buku, buku.penulis, buku.kategori, buku.stok,
        IFNULL(COUNT(peminjaman.id_peminjaman), 0) AS jumlah_peminjaman
    FROM buku LEFT JOIN peminjaman ON buku.id_buku = peminjaman.id_buku GROUP BY buku.id_buku, buku.judul_buku, buku.penulis, buku.kategori, buku.stok;
END $$
DELIMITER ;
```