

# 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

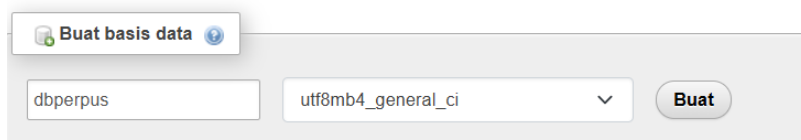
ID Siswa	Nama	Kelas
1	Andi Saputra	X-RPL
2	Budi Wijaya	X-TKJ
3	Citra Lestari	XI-RPL
4	Dewi Kurniawan	XI-TKJ
5	Eko Prasetyo	XII-RPL
6	Farhan Maulana	XII-TKJ
7	Gita Permata	X-RPL
8	Hadi Sucipto	X-TKJ
9	Intan Permadi	XI-RPL
10	Joko Santoso	XI-TKJ
11	Kartika Sari	XII-RPL
12	Lintang Putri	XII-TKJ
13	Muhammad Rizky	X-RPL
14	Novi Andriana	X-TKJ

### 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.



The screenshot shows a web-based database creation tool. At the top, there is a button labeled 'Buat basis data'. Below it, there is a text input field containing 'dbperpus' and a dropdown menu showing 'utf8mb4\_general\_ci'. To the right of these fields is a button labeled 'Buat'.

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, UPDATE, DELETE.

```

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
(1, 2, '2025-02-01', '2025-02-08', 'Dipinjam'),
(2, 5, '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, 1, '2025-01-25', '2025-02-01', 'Dikembalikan');

```

**Data berhasil di input.**

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

**Table 1**

```

DELIMITER //

CREATE PROCEDURE InsertBuku()

BEGIN

    INSERT INTO buku (judul_buku, penulis, kategori, stok) VALUES

    ('Sistem Operasi', 'Dian Kurniawan', 'Teknologi', 6),

    ('Jaringan Komputer', 'Ahmad Fauzi', 'Teknologi', 5),

    ('Cerita Rakyat Nusantara', 'Lestari Dewi', 'Sastra', 9),

    ('Bahasa Inggris untuk Pemula', 'Jane Doe', 'Bahasa', 10),

    ('Biologi Dasar', 'Budi Rahman', 'Sains', 7),

    ('Kimia Organik', 'Siti Aminah', 'Sains', 5),

    ('Teknik Elektro', 'Ridwan Hakim', 'Teknik', 6),

    ('Fisika Modern', 'Albert Einstein', 'Sains', 4),

    ('Manajemen Waktu', 'Steven Covey', 'Pengembangan', 8),

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

END //

```

## Table 2

```

DELIMITER //

CREATE PROCEDURE InsertSiswa()

BEGIN

    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'),

    ('Farhan Maulana', 'XII-TKJ'),

    ('Gita Permata', 'X-RPL'),

    ('Hadi Sucipto', 'X-TKJ'),

    ('Intan Permadi', 'XI-RPL'),

    ('Joko Santoso', 'XI-TKJ');

END //

```

**Table 3**

DELIMITER //

CREATE PROCEDURE Insertpeminjaman(

IN id\_siswa INT,

IN id\_buku INT,

IN Tanggal\_pinjam DATE,

IN Tanggal\_kembali DATE,

IN Status ENUM('Dipinjam', 'Kembalikan')

)

BEGIN

INSERT INTO peminjaman (id\_siswa, id\_buku, tanggal\_pinjam, tanggal\_kembali, status)

VALUES (id\_siswa, id\_buku, Tanggal\_pinjam, Tanggal\_kembali, Status);

END //

CALL Insertpeminjaman(15, 7, '2025-02-01', '2025-02-08', 'Dipinjam');

CALL Insertpeminjaman(7, 1, '2025-01-29', '2025-02-05', 'Dikembalikan');

CALL Insertpeminjaman(8, 9, '2025-02-03', '2025-02-10', 'Dipinjam');

CALL Insertpeminjaman(13, 4, '2025-01-27', '2025-02-03', 'Dikembalikan');

CALL Insertpeminjaman(10, 11, '2025-02-01', '2025-02-08', 'Dipinjam');

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

**Table 1**

Update

DELIMITER //

CREATE PROCEDURE UpdateBuku(

IN Id\_BukuBaru INT,

IN JudulBukuBaru VARCHAR(50),

IN PenulisBaru VARCHAR(20),

IN KategoriBaru VARCHAR(20),

IN StokBaru INT

)

BEGIN

UPDATE buku

SET judul\_buku = JudulBukuBaru,

penulis = PenulisBaru,

kategori = KategoriBaru,

stok = StokBaru

```
WHERE id_buku = Id_BukuBaru;  
END //
```

Delete

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

## Table 2

Update

```
DELIMITER //  
CREATE PROCEDURE UpdateSiswa(  
    IN Id_SiswaBaru INT,  
    IN NamaBaru VARCHAR(50),  
    IN KelasBaru VARCHAR(15)  
)  
BEGIN  
    UPDATE siswa  
    SET nama = NamaBaru,  
        kelas = KelasBaru  
    WHERE id_siswa = Id_SiswaBaru;  
END //
```

Delete

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

## Table 3

Update

```
DELIMITER //  
CREATE PROCEDURE UpdatePeminjaman(  
    IN Id_PeminjamanBaru INT,  
    IN Id_SiswaBaru INT,  
    IN Id_BukuBaru INT,
```

```

        IN TanggalPinjamBaru DATE,
        IN TanggalKembaliBaru DATE,
        IN StatusBaru VARCHAR(15) -- ENUM diganti ke VARCHAR karena nggak bisa di
parameter prosedur
    )
BEGIN
    UPDATE peminjaman
    SET id_siswa = Id_SiswaBaru,
        id_buku = Id_BukuBaru,
        tanggal_pinjam = TanggalPinjamBaru,
        tanggal_kembali = TanggalKembaliBaru,
        status = StatusBaru
    WHERE id_peminjaman = Id_PeminjamanBaru;
END //

```

Delete

```

DELIMITER //
CREATE PROCEDURE DeletePeminjaman(
    IN Id_PeminjamanBaru INT
)
BEGIN
    DELETE FROM peminjaman WHERE id_peminjaman = Id_PeminjamanBaru;
END //

```

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

**Table 1**

```

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

```

**Table 2**

```

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

```

**Table 3**

```

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

```

7. Stok buku pada saat dipinjam berkurang secara otomatis.

DELIMITER //

```
CREATE TRIGGER KurangiStokBuku  
AFTER INSERT ON peminjaman  
FOR EACH ROW
```

```
BEGIN
```

```
UPDATE buku SET stok = stok - 1 WHERE id_buku = NEW.id_buku;
```

```
END //
```

**Cek Triger di tab SQL:**

```
SHOW TRIGGERS FROM dbperpus;
```

8. Stok buku pada saat dikembalikan bertambah secara otomatis.

DELIMITER //

```
CREATE TRIGGER TambahStokBuku  
AFTER UPDATE ON peminjaman  
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 //
```

**Cek Triger di tab SQL:**

```
SHOW TRIGGERS FROM dbperpus;
```

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

DELIMITER //

```
CREATE PROCEDURE KembalikanBuku(  
    IN IdPeminjamanBaru INT  
)
```

```
)  
BEGIN
```

```
    UPDATE peminjaman
```

```
    SET tanggal_kembali = CURDATE(), status = 'Dikembalikan'
```

```
    WHERE id_peminjaman = IdPeminjamanBaru;
```

```
END //
```

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

DELIMITER //



```
CREATE PROCEDURE DaftarSiswaPeminjam()
BEGIN
    SELECT DISTINCT siswa.id_siswa, siswa.nama, siswa.kelas
    FROM siswa
    JOIN peminjaman ON siswa.id_siswa = peminjaman.id_siswa
    WHERE peminjaman.status IN ('Dipinjam', 'Dikembalikan');
END //
```

**Memanggil SQL :**

```
CALL DaftarSiswaPeminjam();
```

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

```
DELIMITER //
```

```
CREATE PROCEDURE DaftarSiswaLengkap()
BEGIN
    SELECT siswa.id_siswa, siswa.nama, siswa.kelas
    FROM siswa
    LEFT JOIN peminjaman ON siswa.id_siswa = peminjaman.id_siswa
    WHERE peminjaman.id_siswa IS NULL OR peminjaman.status IN ('Dipinjam',
'Dikembalikan');
END //
```

**Memanggil SQL :**

```
CALL DaftarSiswaLengkap();
```

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

```
DELIMITER //
```

```
CREATE PROCEDURE DaftarBukuLengkap()
BEGIN
    SELECT buku.id_buku, buku.judul_buku, buku.penulis, buku.kategori, buku.stok
    FROM buku
    LEFT JOIN peminjaman ON buku.id_buku = peminjaman.id_buku
    WHERE peminjaman.id_buku IS NULL OR peminjaman.status IN ('Dipinjam',
'Dikembalikan');
END //
```

**Memanggil SQL :**

```
CALL DaftarBukuLengkap();
```

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