

Nama : Anne Dwi Revina

Kelas : XI RPL 2

No. Absen : 03

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

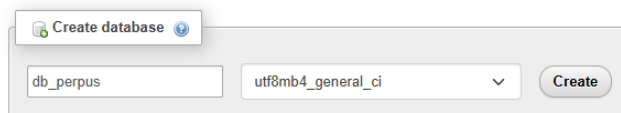
3. Tabel Peminjaman

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.

Databases



The screenshot shows a web-based database creation tool. At the top, there is a 'Create database' button with a plus icon. Below it, there is a text input field containing 'db_perpus' and a dropdown menu showing 'utf8mb4_general_ci'. To the right of these fields is a 'Create' button.

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');

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

Insert Buku :

DELIMITER //

CREATE PROCEDURE InsertBuku(

IN judul_buku varchar(225),

IN penulis varchar(225),

IN kategori varchar(100),

IN stok int

)

BEGIN

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

(judul_buku, penulis, kategori, stok);

END //

DELIMITER ;

Insert Siswa :

DELIMITER //

CREATE PROCEDURE InsertSiswa2(

IN nama varchar(50),

IN kelas varchar(50)

```
)  
BEGIN  
    INSERT INTO siswa (nama, kelas) VALUES  
    (nama, kelas);
```

```
END //  
DELIMITER ;
```

Insert Peminjaman :

```
DELIMITER //  
CREATE PROCEDURE InsertPeminjaman2(  
    IN Id_siswa int,  
    IN Id_buku int,  
    IN Tanggal_pinjam date,  
    IN Tanggal_kembali date,  
    IN Status ENUM('Dipinjam','Dikembalikan')  
)  
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 InsertBuku('Jaringan Komputer', 'Ahmad Fauzi', 'Teknologi', 5);
```

```
CALL InsertSiswa('Olivia Hernanda', 'XI-RPL');
```

```
CALL InsertPeminjaman(6, 7, '2025-02-05', '2025-02-12', 'Dipinjam');
```

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

Update tabel buku :

```
DELIMITER //
```

```
CREATE PROCEDURE UpdateBuku(  
    IN p_id_buku INT,  
    IN p_judul_buku VARCHAR(255),  
    IN p_penulis VARCHAR(255),  
    IN p_kategori VARCHAR(100),  
    IN p_stok INT  
)
```

```
BEGIN
```

```
    UPDATE buku
```

```
    SET judul_buku = p_judul_buku,
```

```
        penulis = p_penulis,
```

```
        kategori = p_kategori,
```

```
        stok = p_stok
```

```
    WHERE id_buku = p_id_buku;
```

```
END //
```

```
DELIMITER ;
```

Delete tabel buku :

```
DELIMITER //
```

```
CREATE PROCEDURE DeleteBuku(  
    IN p_id_buku INT  
)
```

```
    IN p_id_buku INT  
)
```

```
)
```

```
BEGIN
```

```
    DELETE FROM buku WHERE id_buku = p_id_buku;
```

```
END //
```

```
DELIMITER ;
```

Update tabel siswa :

```
DELIMITER //
```

```
CREATE PROCEDURE UpdateSiswa(  
    IN p_id_siswa INT,  
    IN p_nama VARCHAR(255),  
    IN p_kelas VARCHAR(50)
```

```
    IN p_id_siswa INT,
```

```
    IN p_nama VARCHAR(255),
```

```
    IN p_kelas VARCHAR(50)
```

)

BEGIN

UPDATE siswa

SET nama = p_nama,

kelas = p_kelas

WHERE id_siswa = p_id_siswa;

END //

DELIMITER ;

Delete tabel siswa :

DELIMITER //

CREATE PROCEDURE DeleteSiswa(

IN p_id_siswa INT

)

BEGIN

DELETE FROM siswa WHERE id_siswa = p_id_siswa;

END //

DELIMITER ;

Update tabel peminjaman :

DELIMITER //

CREATE PROCEDURE UpdatePeminjaman(

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

UPDATE peminjaman

SET id_siswa = p_id_siswa,

id_buku = p_id_buku,

```
    tanggal_pinjam = p_tanggal_pinjam,  
    tanggal_kembali = p_tanggal_kembali,  
    status = p_status  
WHERE id_peminjaman = p_id_peminjaman;  
END //  
DELIMITER ;
```

Delete tabel peminjaman :

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

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

Table buku :

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

Tabel siswa :

```
DELIMITER //  
CREATE PROCEDURE GetAllSiswa()  
BEGIN  
    SELECT * FROM siswa;
```



```
END //
```

```
DELIMITER ;
```

Tabel peminjaman :

```
DELIMITER //
```

```
CREATE PROCEDURE GetAllPeminjaman()
```

```
BEGIN
```

```
    SELECT * FROM peminjaman;
```

```
END //
```

```
DELIMITER ;
```

```
CALL GetAllBuku();
```

```
CALL GetAllSiswa();
```

```
CALL GetAllPeminjaman();
```

7. Stok buku pada saat dipinjam berkurang secara otomatis.

```
DELIMITER //
```

```
CREATE PROCEDURE PinjamBuku(
```

```
    IN p_id_siswa INT,
```

```
    IN p_id_buku INT,
```

```
    IN p_tanggal_pinjam DATE
```

```
)
```

```
BEGIN
```

```
-- Insert record peminjaman dengan status 'Dipinjam' dan tanggal_kembali NULL
```

```
INSERT INTO peminjaman (id_siswa, id_buku, tanggal_pinjam, tanggal_kembali, status)
```

```
VALUES (p_id_siswa, p_id_buku, p_tanggal_pinjam, NULL, 'Dipinjam');
```

```
-- Kurangi stok buku
```

```
UPDATE buku SET stok = stok - 1 WHERE id_buku = p_id_buku;
```

```
END //
```

```
DELIMITER ;
```

```
CALL PinjamBuku(2, 3, '2025-02-15');
```

8. Stok buku pada saat dikembalikan bertambah secara otomatis.

```
DELIMITER //
```

```
CREATE PROCEDURE KembalikanBuku(
```

```
    IN p_id_peminjaman INT,
```

```
    IN p_id_buku INT
```

```
)
```

```
BEGIN
```

```
-- Update status peminjaman dan set tanggal_kembali dengan CURRENT_DATE
```

```
UPDATE peminjaman
```

```
SET status = 'Dikembalikan', tanggal_kembali = CURRENT_DATE
```

```
WHERE id_peminjaman = p_id_peminjaman;
```

```
-- Tambah stok buku
```

```
UPDATE buku SET stok = stok + 1 WHERE id_buku = p_id_buku;
```

```
END //
```

```
DELIMITER ;
```

```
CALL KembalikanBuku(7, 3);
```

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

```
DELIMITER //
```

```
CREATE PROCEDURE ReturnBook(
```

```
    IN p_id_peminjaman INT,
```

```
    IN p_id_buku INT
```

```
)
```

BEGIN

-- Update status peminjaman menjadi 'Dikembalikan' dan set tanggal_kembali dengan CURRENT_DATE()

UPDATE peminjaman

SET status = 'Dikembalikan',

tanggal_kembali = CURRENT_DATE()

WHERE id_peminjaman = p_id_peminjaman;

-- Tambahkan stok buku karena buku sudah dikembalikan

UPDATE buku

SET stok = stok + 1

WHERE id_buku = p_id_buku;

END //

DELIMITER ;

CALL ReturnBook(1, 2);

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

DELIMITER //

CREATE PROCEDURE GetSiswaPernahPinjam()

BEGIN

SELECT DISTINCT s.id_siswa, s.nama, s.kelas

FROM siswa s

JOIN peminjaman p ON s.id_siswa = p.id_siswa;

END //

DELIMITER ;

CALL GetSiswaPernahPinjam();

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

DELIMITER //

```

CREATE PROCEDURE GetSemuaSiswa()
BEGIN
    SELECT s.id_siswa, s.nama, s.kelas,
           CASE
               WHEN p.id_peminjaman IS NULL THEN 'Belum Pernah Meminjam'
               ELSE 'Pernah Meminjam'
           END AS status_peminjaman
    FROM siswa s
    LEFT JOIN peminjaman p ON s.id_siswa = p.id_siswa;
END //
DELIMITER ;

CALL GetSemuaSiswa();

```

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

```

DELIMITER //
CREATE PROCEDURE GetSemuaBuku()
BEGIN
    SELECT b.id_buku, b.judul_buku, b.penulis, b.kategori, b.stok,
           CASE
               WHEN p.id_peminjaman IS NULL THEN 'Belum Pernah Dipinjam'
               ELSE 'Pernah Dipinjam'
           END AS status_peminjaman
    FROM buku b
    LEFT JOIN peminjaman p ON b.id_buku = p.id_buku;
END //
DELIMITER ;

CALL GetSemuaBuku();

```