

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 perpustakaan;
2. Buatlah table buku, siswa dan peminjaman.
Table Buku:
create table buku(id_buku int primary key,judul varchar(255),penulis varchar(255),kategori varchar(50),stok int);
Table Siswa:
create table siswa(id_siswa int primary key,nama varchar(50),kelas varchar(50));
Table Peminjaman:
create table pinjaman(id_pinjaman int primary key,id_siswa int,id_buku int,tgl_pinjam date,tgl_kembali date,status varchar(50));
3. Input 5 record di setiap table menggunakan query INSERT, UPDATE, DELETE.
INSERT
insert into buku(id_buku,judul,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','Jhon Smith','Sejarah',3),
-> (5,'Pemrograman Web Dengan PHP','Eko Prasetyo','Teknologi',8);
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 Kumiawan','XI-TKJ'),
-> (5,'Eko Prasetyo','XII-RPL');

```

insert into pinjaman (id_pinjaman,id_siswa,id_buku,tgl_pinjam,tgl_kembali,status)values
-> (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');

```

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

Table Buku

DELIMITER //

CREATE PROCEDURE insertbuku(

IN Id_buku int,

IN Judul varchar(255),

IN Penulis varchar(255),

IN Kategori varchar(50),

IN Stok int

)

BEGIN

INSERT INTO buku(id_buku,judul,penulis,kategori,stok)

VALUES (Id_buku,Judul,Penulis,Kategori,Stok);

END //

CALL insertbuku (6,'Sistem Operasi','Dian Kurniawan','Teknologi',6);

CALL insertbuku (7,'Jaringan Komputer','Ahmad Fauzi','Teknologi',5);

CALL insertbuku(8,'Cerita Rakyat Nusantara' , 'Lestari Dewi','Sastra',9);

CALL insertbuku(9,'Bahasa Inggris untuk Pemula' , 'Jane Doe','Bahasa',10);

CALL insertbuku(10,'Biologi Dasar','Budi Rahman','Sains',7);

CALL insertbuku(11,'Kimia Organik', 'Siti Aminah','Sains',5);

CALL insertbuku(12,'Teknik Elektro','Ridwan Hakim', 'Teknik',6);

CALL insertbuku(13, 'Fisika Modern','Albert Einstein','Sains',4);

CALL insertbuku(14,'Manajemen Waktu','Steven Covey','Pengembangan', 8);

CALL insertbuku(15,'Strategi Belajar Efektif','Tony Buzan','Pendidikan',6);

Table pinjaman

DELIMITER //

CREATE PROCEDURE insertpinjam(

IN Id_pinjaman int,

IN Id_siswa int,

IN Id_buku int,

IN Tgl_pinjam date,

IN Tgl_kembali date,

IN Statu varchar(50)

)

BEGIN

INSERT INTO buku(id_pinjaman,id_siswa,id_buku,tgl_pinjam,tgl_kembali,status)

VALUES (Id_pinjaman,Id_siswa,Id_buku,Tgl_pinjam,Tgl_kembali,Statu);

END //

CALL insertpinjam(6,15,7,'2025-02-01','2025-02-08','Dipinjam');

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

```
CALL insertpinjam(8,8,9,'2025-02-03','2025-02-10','Dipinjam');
CALL insertpinjam(9,13,4,'2025-01-27','2025-02-03','Dikembalikan');
CALL insertpinjam(10,10,11,'2025-02-01','2025-02-08','Dipinjam');
```

Table Siswa

DELIMITER //

```
CREATE PROCEDURE insertbuku(
```

```
    IN Id_siswa int,
```

```
    IN Nama varchar(50),
```

```
    IN Kelas varchar(50),
```

```
)
```

```
BEGIN
```

```
    INSERT INTO siswa(id_siswa,nama,kelas)
```

```
    VALUES (Id_siswa,Nama,Kelas);
```

```
END //
```

```
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');
```

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

Table buku:

DELIMITER //

```
CREATE PROCEDURE updatebuku(
```

```
    IN Id_buku int,
```

```
    IN Judul varchar(255),
```

```
    IN Penulis varchar(255),
```

```
    IN Kategori varchar(50),
```

```
    IN Stok int
```

```
)
```

```
BEGIN
```

```
UPDATE buku SET judul=Judul,penulis=Penulis,kategori=Kategori,stok=Stok
```

```
WHERE id_buku=Id_buku;
```

```
END //
```

DELIMITER //

```
CREATE PROCEDURE deletebukur(
```

```
    IN id INT
```

```
)
```

```
BEGIN
```

```

DELETE FROM buku
WHERE id_buku=id;
END //
Table pinjaman:
DELIMITER //
CREATE PROCEDURE updatepinjaman(
    IN Id_pinjam int,
    IN Id_siswa int,
    IN Id_buku int,
    IN tgl_pin date,
    IN tgl_kem date,
    IN sts varchar(50)
)
BEGIN
UPDATE      buku
SET
id_siswa=Id_siswa,id_buku=Id_buku,tgl_pinjam=tgl_pin,tgl_kembali=tgl_kem,status=sts
WHERE id_pinjaman=Id_pinjam;

END //

DELIMITER //

CREATE PROCEDURE deletepinjam(
    IN id INT
)
BEGIN
DELETE FROM pinjaman
WHERE id_pinjaman=id;
END //
Table siswa:
DELIMITER //
CREATE PROCEDURE updatesiswa(
    IN Id_siswa int,
    IN Nama varchar(50),
    IN Kelas varchar(50)
)
BEGIN
UPDATE siswa SET nama=Nama,kelas=Kelas
WHERE id_siswa=Id_siswa;

END //

DELIMITER //

CREATE PROCEDURE deletesiswa(

```

```

        IN id INT
    )
BEGIN
    DELETE FROM siswa
    WHERE id_siswa=id;
END //

```

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

Table Buku:

DELIMITER //

```

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

```

CALL selectbuku();

Table Pinjaman:

DELIMITER //

```

CREATE PROCEDURE selectpinjam()
BEGIN
    SELECT * FROM pinjaman;
END //

```

CALL selectpinjam();

Table Siswa:

DELIMITER //

```

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

```

CALL selectsiswa();

7. Stok buku pada saat dipinjam berkurang secara otomatis.

DELIMITER //

```

CREATE TRIGGER update_stok_buku_after_pinjam
AFTER INSERT ON peminjaman
FOR EACH ROW
BEGIN

```

```

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

```

8. Stok buku pada saat dikembalikan bertambah secara otomatis.

DELIMITER//

```

CREATE TRIGGER update_stok_buku_after_kembali

```

```

AFTER UPDATE ON pinjaman
FOR EACH ROW
BEGIN
    IF OLD.tgl_kembali IS NULL AND NEW.tgl_kembali IS NOT NULL THEN
        UPDATE buku
        SET stok = stok + 1
        WHERE id_buku = NEW.id_buku;
    END IF;
END//

```

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

```

CREATE PROCEDURE kembalikan_buku(IN Id_peminjaman INT)
BEGIN
    UPDATE peminjaman
    SET tanggal_kembali = CURRENT_DATE
    WHERE id_pinjaman = Id_peminjaman;
END //

```

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

```

CREATE PROCEDURE daftar_siswa_meminjam()
BEGIN
    SELECT DISTINCT siswa.id_siswa, siswa.nama, siswa.kelas
    FROM siswa
    JOIN pinjaman ON siswa.id_siswa = pinjaman.id_siswa;
END //

```

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

```

CREATE PROCEDURE semua_siswa_termasuk_tidak_meminjam()
BEGIN
    SELECT siswa.id_siswa, siswa.nama, siswa.kelas
    FROM siswa
    LEFT JOIN pinjaman ON siswa.id_siswa = pinjaman.id_siswa;
END //

```

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

```

CREATE PROCEDURE semua_buku_termasuk_tidak_dipinjam()
BEGIN
    SELECT buku.id_buku, buku.judul, buku.penulis, buku.kategori, buku.stok
    FROM buku
    LEFT JOIN pinjaman ON buku.id_buku = pinjaman.id_buku;
END //

```

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