

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	John Smith	3
5	Pemrograman Web dengan PHP	Eko Prasetyo	Eko Prasetyo	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.
2. Buatlah table buku, siswa dan peminjaman.
3. Input 5 record di setiap table menggunakan query INSERT.
4. Input 10 record di setiap table menggunakan stored procedure INSERT.
5. Buatlah stored procedure UPDATE, DELETE di setiap table.
6. Buatlah stored procedure untuk menampilkan seluruh record di setiap table.
7. Stok buku pada saat dipinjam berkurang secara otomatis.
8. Stok buku pada saat dikembalikan bertambah secara otomatis.
9. Buatlah stored procedure untuk mengembalikan buku dan gunakan tanggal pengembalian sesuai dengan tanggal saat mengembalikan (CURRENT DATE).
10. Buatlah stored procedure untuk menampilkan daftar siswa yang pernah meminjam buku.
11. Buatlah stored procedure untuk menampilkan semua siswa, termasuk yang tidak pernah meminjam buku.
12. Buatlah stored procedure untuk menampilkan semua buku, termasuk yang belum pernah dipinjam.

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

Jawaban

1. `1 CREATE DATABASE db_perpus;`

2.

```
CREATE TABLE Siswa(  
    IDiswa int primary key AUTO_INCREMENT,  
    Nama varchar(50),  
    Kelas varchar(50));|
```

Anda memiliki kuota 1000000 karakter

```
1 CREATE TABLE buku(  
2 IDBuku int primary key AUTO_INCREMENT,  
3 JudulBuku varchar(50),  
4 Penulis varchar(50),  
5 Kategori varchar(50),  
6 Stok int);|
```

```
CREATE TABLE Peminjaman(  
    IDPeminjaman int primary key AUTO_INCREMENT,  
    IDiswa int,  
    IDBuku int,  
    TanggalPinjam DATE,  
    TanggalKembali Date,  
    Status varchar(50));|
```

3.

```
1 INSERT INTO Buku(JudulBuku, Penulis, Kategori, Stok) VALUES(  
2 "Algoritma dan Pemrograman", "Andi Wijaya", "Teknologi", 5),  
3 ("Dasar-dasar Database", "Budi Santoso", "Teknologi", 7),  
4 ("Matematika Diskrit", "Rina Sari", "Matematika", 4),  
5 ("Sejarah Dunia", "John Smith", "John Smith", 3),  
6 ("Pemrograman Web dengan PHP", "Eko Prasetyo", "Eko Prasetyo", 8);
```

```
1 INSERT INTO Peminjaman(IDiswa, IDBuku, TanggalPinjam, TanggalKembali, Status) VALUES(  
2 11, 2, "2025-02-01", "2025-02-08", "Dipinjam"),  
3 (2, 5, "2025-01-28", "2025-02-04", "Dikembalikan"),  
4 (3, 8, "2025-02-02", "2025-02-09", "Dipinjam"),  
5 (4, 10, "2025-01-30", "2025-02-06", "Dikembalikan"),  
6 (5, 3, "2025-01-25", "2025-02-01", "Dikembalikan");
```

```

1 INSERT INTO Siswa(Nama, Kelas) VALUES(
2     "Andi Saputra", "X-RPL"),
3     ("Budi Wijaya", "X-TKJ"),
4     ("Citra Lestari", "XI-RPL"),
5     ("Dewi Kurniawan", "XI-TKJ"),
6     ("Eko Prasetyo", "XII-RPL");|

```

4.

```

1 DELIMITER //
2 CREATE PROCEDURE insert_buku(
3     pJudul varchar(50),
4     pPenulis varchar(50),
5     pKategori varchar(50),
6     pStok int)
7 BEGIN
8     INSERT INTO buku(JudulBuku, Penulis, Kategori, stok) VALUES(pJudul, pPenulis, pKategori, pStok);|
9 END //
10 DELIMITER ;

```

```

1 DELIMITER //
2 CREATE PROCEDURE insert_siswa(
3     pNama varchar(50),
4     pKelas varchar(50))
5 BEGIN
6     INSERT INTO siswa (Nama, Kelas) VALUES(pNama, pKelas);
7 END //
8 DELIMITER ;|

```

```

1 DELIMITER //
2 CREATE PROCEDURE insert_peminjaman(
3     pIDSiswa int,
4     pIDBuku int,
5     pTanggalPinjam varchar(50),
6     pTanggalKembali varchar(50),
7     pStatus varchar(50))
8 BEGIN
9     INSERT INTO peminjaman(IDSiswa, IDBuku, TanggalPinjam, TanggalKembali, Status) VALUES(pIDSiswa, pIDBuku, pTanggalPinjam,
10     pTanggalKembali, pStatus);
11 END //
12 DELIMITER ;

```

Input data

Insert data ke table buku :

```

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

```

Insert data ke table siswa :

```

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

```

5. Procedure update delete table buku

```

1 DELIMITER //
2 CREATE PROCEDURE update_buku(
3     pIDBuku INT,
4     pJudulBuku VARCHAR(50),
5     pPengarangBuku VARCHAR(50),
6     pKategoriBuku VARCHAR(50),
7     pStok INT)
8 BEGIN
9     UPDATE buku
10    SET judul = pJudulBuku, pengarang = pPengarangBuku, kategori = pKategoriBuku, stok = pStokBuku
11    WHERE id = pIdBuku;
12 END //
13 DELIMITER ;

```

```

1 DELIMITER //
2 CREATE PROCEDURE delete_buku(
3     pIdBuku INT)
4 BEGIN
5     DELETE FROM buku WHERE id = pIdBuku;
6 END //
7 DELIMITER ;

```

Procedure update delete table siswa

```

1 DELIMITER //
2 CREATE PROCEDURE update_siswa(
3     pIDSiswa INT,
4     pNamaSiswa VARCHAR(50),
5     pKelasSiswa VARCHAR(50))
6 BEGIN
7     UPDATE siswa
8     SET nama = pNamaSiswa, kelas = pKelasSiswa
9     WHERE id = pIDSiswa;
10 END //
11 DELIMITER ;

```

```

1 DELIMITER $$
2 CREATE PROCEDURE delete_siswa(
3     pIDSiswa int
4 )
5 BEGIN
6     DELETE FROM siswa WHERE id = pIDSiswa;
7 END $$
8 DELIMITER ;

```

Procedure update delete table peminjaman

```

1 DELIMITER //
2 CREATE PROCEDURE update_peminjaman(
3     pIdBuku INT,
4     pIdSiswa INT,
5     pTanggalPinjam DATE,
6     pTanggalKembali DATE,
7     pStatusPinjam VARCHAR(50))
8 BEGIN
9     UPDATE peminjaman
10    SET IDBuku = pIDBuku, IDSiswa = pIDSiswa, TanggalPinjam = pTanggalPinjam, TanggalKembali = pTanggalKembali, status = pStatus
11    WHERE id = IDPeminjaman;
12 END //
13
14 DELIMITER ;

```

```

1 DELIMITER //
2 CREATE PROCEDURE delete_peminjaman(
3     pIdPeminjaman INT)
4 BEGIN
5     DELETE FROM peminjaman WHERE id = pIDPeminjaman;
6 END //
7 DELIMITER ;

```

6. Membuat procedure untuk menampilkan semua data

```

1 DELIMITER //
2 CREATE PROCEDURE tampilkan_buku()
3 BEGIN
4     SELECT * FROM buku;
5 END //
6 DELIMITER ;

```

```

1 DELIMITER //
2 CREATE PROCEDURE tampilkan_siswa()
3 BEGIN
4     SELECT * FROM siswa;
5 END //
6 DELIMITER ;

```

```

1 DELIMITER //
2 CREATE PROCEDURE tampilkan_peminjaman()
3 BEGIN
4     SELECT * FROM peminjaman;
5 END //
6 DELIMITER ;

```

7. Membuat trigger agar stok buku yang dipinjam berkurang otomatis

```

1 DELIMITER //
2 CREATE TRIGGER after_insert_peminjaman
3 AFTER INSERT ON peminjaman
4 FOR EACH ROW
5 BEGIN
6     UPDATE buku
7     SET Stok = Stok - 1
8     WHERE IDBuku = NEW.IDBuku;
9 END //
10 DELIMITER ;

```

8. Trigger agar stok buku agar kembali bertambah saat di kembalikan

```

1 DELIMITER //
2 CREATE TRIGGER after_update_peminjaman
3 AFTER UPDATE ON peminjaman
4 FOR EACH ROW
5 BEGIN
6     IF NEW.Status = 'Dikembalikan'
7     THEN UPDATE buku SET Stok = Stok + 1
8     WHERE IDBuku = NEW.IDBuku;
9     END IF;
10 END //
11 DELIMITER ;

```

9. Procedure untuk mengembalikan buku dan memakai tanggal sekarang

```

1 DELIMITER //
2 CREATE PROCEDURE kembalikanbuku(
3     IDPeminjaman int)
4 BEGIN
5     UPDATE peminjaman
6     SET Status = 'Dikembalikan', Tanggal_Kembali = CURRENT_DATE
7     WHERE ID_Peminjaman = IDPeminjaman;
8 END //
9 DELIMITER ;

```

10. Procedure untuk menampilkan semua siswa yg pernah meminjam buku

```

1 DELIMITER //
2 CREATE PROCEDURE SiswaPernahMinjam()
3 BEGIN
4     SELECT DISTINCT s.ID_Siswa, s>Nama, s.Kelas
5     FROM siswa s
6     INNER JOIN peminjaman p ON s.ID_Siswa = p.ID_Siswa;
7 END //
8 DELIMITER ;

```

11. Procedure untuk menampilkan semua siswa (termasuk yg belum pernah meminjam)


```

1 DELIMITER //
2 CREATE PROCEDURE SemuaSiswa()
3 BEGIN
4     SELECT s.ID_Siswa, s>Nama, s.Kelas,
5     CASE
6         WHEN p.ID_Siswa IS NULL THEN 'Belum Pernah Meminjam'
7         ELSE 'Pernah Meminjam'
8     END AS Status
9     FROM siswa s
10    LEFT JOIN peminjaman p ON s.ID_Siswa = p.ID_Siswa
11    GROUP BY s.ID_Siswa;
12 END //
13 DELIMITER ;|

```

12. Procedure untuk menampilkan semua buku (termasuk yg belum pernah d pinjam)

```

1 DELIMITER //
2 CREATE PROCEDURE SemuaBuku()
3 BEGIN
4     SELECT b.ID_Buku, b.Judul_Buku, b.Penulis, b.Kategori, b.Stok,
5     CASE
6         WHEN p.IDBuku IS NULL THEN 'Belum Pernah Dipinjam'
7         ELSE 'Pernah Dipinjam'
8     END AS Status
9     FROM buku b
10    LEFT JOIN peminjaman p ON b.ID_Buku = p.IDBuku
11    GROUP BY b.ID_Buku;
12 END //
13 DELIMITER ;|

```