

SISTEM PERPUSTAKAAN SEKOLAH

Nama : Zahra Tusita

Kelas : 11 RPL 1

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
	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

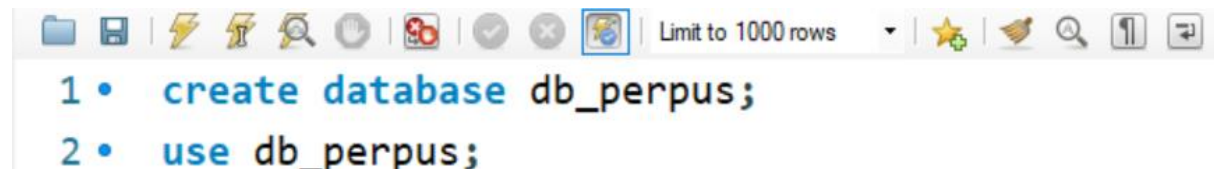
3. Tabel Peminjaman

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.

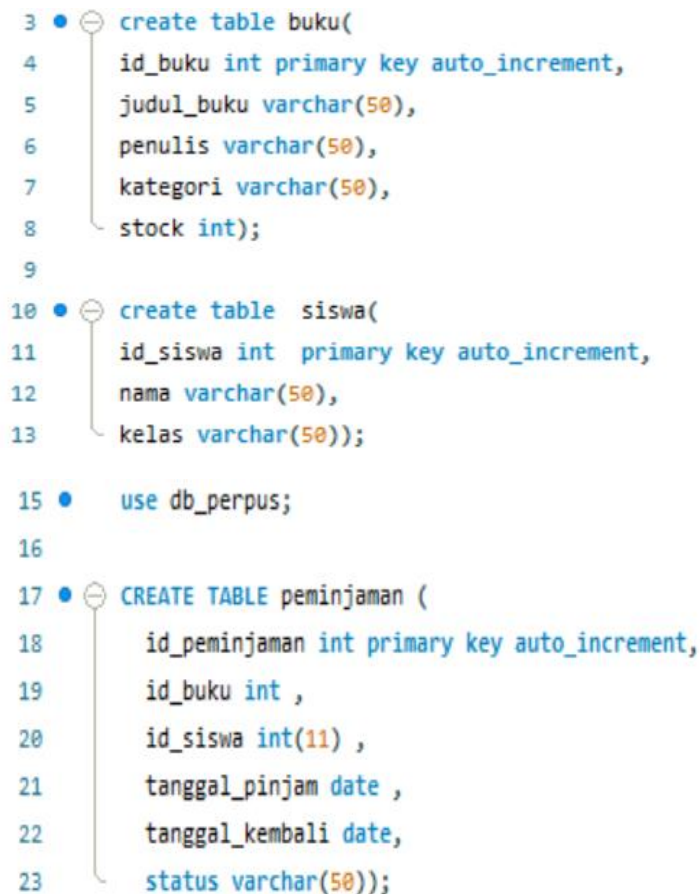


```

1 • create database db_perpus;
2 • use db_perpus;

```

2. Buatlah table buku, siswa dan peminjaman.



```

3 • create table buku(
4     id_buku int primary key auto_increment,
5     judul_buku varchar(50),
6     penulis varchar(50),
7     kategori varchar(50),
8     stock int);
9
10 • create table siswa(
11     id_siswa int primary key auto_increment,
12     nama varchar(50),
13     kelas varchar(50));
14
15 • use db_perpus;
16
17 • CREATE TABLE peminjaman (
18     id_peminjaman int primary key auto_increment,
19     id_buku int ,
20     id_siswa int(11) ,
21     tanggal_pinjam date ,
22     tanggal_kembali date,
23     status varchar(50));

```

3. Input 5 record di setiap table menggunakan query INSERT. **Tabel Siswa**

```

38 ● INSERT INTO siswa(nama, kelas) VALUES
39     ('Andi Saputra', 'X-RPL'),
40     ('Budi Wijaya', 'X-TKJ'),
41     ('Citra Lestari', 'XI-RPL'),
42     ('Dewi Kurniawan', 'XI-TKJ'),
43     ('Eko Prasetyo', 'XII-RPL');

```

Tabel Buku

```

27 ● INSERT INTO buku (judul_buku, penulis, kategori, stock) VALUES
28     ('Algoritma dan Pemrograman', 'Andi Wijaya', 'Teknologi', 5);
29
30 ● INSERT INTO buku (judul_buku, penulis, kategori, stock) VALUES
31     ('Dasar-dasar Database', 'Budi Santoso', 'Teknologi', 7),
32     ('Matematika Diskrit', 'Rina Sari', 'Matematika', 4),
33     ('Sejarah Dunia', 'John Smith', 'Sejarah', 3),
34     ('Pemograman', 'Eko Prasetyo', 'Teknologi', 8);

```

Tabel Peminjaman

```

45 ● INSERT INTO peminjaman (id_siswa, id_buku, tanggal_pinjam, tanggal_kembali, status) VALUES
46     (11, 2, '2025-02-01', '2025-02-08', 'Dipinjam'),
47     (2, 5, '2025-01-28', '2025-02-04', 'Dikembalikan'),
48     (3, 8, '2025-02-02', '2025-02-09', 'Dipinjam'),
49     (4, 10, '2025-01-30', '2025-02-06', 'Dikembalikan'),
50     (5, 3, '2025-01-25', '2025-02-01', 'Dikembalikan');

```

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

```

103 ● CREATE PROCEDURE insert_siswa(
104     pNama VARCHAR(100),
105     pKelas VARCHAR(10)
106 )
107 BEGIN
108     INSERT INTO siswa (nama, kelas)
109     VALUES (pNama, pKelas);
110 END $$
111 DELIMITER ;
112
113 ● CALL insert_siswa('Farhan Maulana', 'XII-TKJ');
114 ● CALL insert_siswa('Gita Permata', 'X-RPL');
115 ● CALL insert_siswa('Hadi Sucipto', 'X-TKJ');
116 ● CALL insert_siswa('Intan Permadi', 'XI-RPL');
117 ● CALL insert_siswa('Joko Santoso', 'XI-TKJ');

```

```

118 ● CALL insert_siswa('Kartika Sari', 'XII-RPL');
119 ● CALL insert_siswa('Lintang Putri', 'XII-TKJ');
120 ● CALL insert_siswa('Muhammad Rizky', 'X-RPL');
121 ● CALL insert_siswa('Novi Andriana', 'X-TKJ');
122 ● CALL insert_siswa('Olivia Hernanda', 'XI-RPL');
123

```

Hasil:

Result Grid			
Filter Rows:			
	id_siswa	nama	kelas
▶	1	Andi Saputra	X-RPL
	2	Budi Wijaya	X-TKJ
	3	Hadi Saputra	XI-TKJ
	4	Dewi Kurniawan	XI-TKJ
	5	Eko Prasetyo	XII-RPL
	6	Farhan Maulana	XII-TKJ
	7	Gita Permata	X-RPL

Tabel Buku






```

pPenulis varchar(50),
pKategori varchar(50),
pStok int)
BEGIN
INSERT INTO buku (judul_buku,pennulis,kategori,stock) values (pJudulBuku,pPenulis,pKategori,pStok)
END //

use db_perpus;
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);

```

Hasil:

Result Grid					
Filter Rows: <input type="text"/>					
Edit:   					
Export/Import:  					
	id_buku	judul_buku	penulis	kategori	stock
▶	1	Algoritma dan Pemrograman	Andi Wijaya	Teknologi	5
	2	Dasar-dasar Database Revisi	Budi Santoso	Teknologi	10
	3	Matematika Diskrit	Rina Sari	Matematika	5
	4	Sejarah Dunia	John Smith	Sejarah	3
	5	Pemrograman	Eko Prasetyo	Teknologi	7
	6	Sistem Operasi	Dian Kurniawan	Teknologi	6
	7	Jaringan Komputer	Ahmad Fauzi	Teknologi	5
	8	Cerita Rakyat Nusantara	Lestari Dewi	Sastra	10

buku 19 x

Tabel Peminjaman

```

96 DELIMITER $$
97 • create procedure insert_peminjaman(
98   pId_Siswa int,
99   pId_Buku int,
100  pTanggal_Pinjam date,
101  pTanggal_Kembali date,
102  pStatus varchar(50))
103 • BEGIN
104   INSERT INTO peminjaman (id_siswa,id_buku,tanggal_pinjam,tanggal_kembali,status) values (pid_Siswa,pId_Buku,pTanggal_Pinjam,pTanggal_Kembali,pStatus);
105   END $$
106 DELIMITER ;

108 • CALL insert_peminjaman(15,7,"2025-02-01","2025-02-08","Dipinjam");
109 • CALL insert_peminjaman(7,1,"2025-01-29","2025-02-05","Dikembalikan");
110 • CALL insert_peminjaman(8,9,"2025-02-03","2025-02-10","Dipinjam");
111 • CALL insert_peminjaman(13,4,"2025-01-27","2025-02-03","Dikembalikan");
112 • CALL insert_peminjaman(10,11,"2025-02-01","2025-02-08","Dipinjam");

```

Hasil

	id_peminjaman	id_buku	id_siswa	tanggal_pinjam	tanggal_kembali	status
▶	1	2	1	2025-02-01	2025-02-08	Dipinjam
	2	5	2	2025-01-28	2025-02-04	Dikembalikan
	3	8	3	2025-02-02	2025-02-09	Dipinjam
	4	10	4	2025-01-30	2025-02-06	Dikembalikan
	5	3	5	2025-01-25	2025-02-01	Dikembalikan
	6	2	11	2025-02-01	2025-02-08	Dipinjam
	7	5	2	2025-01-28	2025-02-04	Dikembalikan
	8	8	3	2025-02-02	2025-02-09	Dipinjam
	9	10	4	2025-01-30	2025-02-06	Dikembalikan

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

Update di Tabel Siswa untuk mengupdate nama

```
DELIMITER $$
CREATE PROCEDURE update_siswa(
    pIdSiswa INT,
    pNama VARCHAR(50),
    pKelas VARCHAR(10)
)
BEGIN
    UPDATE siswa
    SET nama = pNama,
        kelas = pKelas
    WHERE id_siswa = pIdSiswa;
END $$
DELIMITER ;

SET SQL_SAFE_UPDATES = 0;

CALL update_siswa(3, 'Hadi Saputra', 'XI-TKJ');
```

Hasil

	id_siswa	nama	kelas
▶	1	Andi Saputra	X-RPL
	2	Budi Wijaya	X-TKJ
	3	Hadi Saputra	XI-TKJ
	4	Dewi Kurniawan	XI-TKJ
	5	Eko Prasetyo	XII-RPL

siswa 20 ✕

Update di Tabel Buku untuk mengupdate stok


```

DELIMITER $$
● CREATE PROCEDURE update_buku(
    pIdBuku INT,
    pJudulBuku VARCHAR(100),
    pPenulis VARCHAR(100),
    pKategori VARCHAR(50),
    pstok INT
)
BEGIN
    UPDATE buku
    SET judul_buku = pJudulBuku,
        penulis = pPenulis,
        kategori = pKategori,
        stock = pstok
    WHERE id_buku = pIdBuku;
END $$
DELIMITER ;
● CALL update_buku(2, 'Dasar-dasar Database Revisi', 'Budi Santoso', 'Teknologi', 10);

```

Hasil

Result Grid					
Filter Rows:					
	id_buku	judul_buku	penulis	kategori	stock
▶	1	Algoritma dan Pemrograman	Andi Wijaya	Teknologi	5
	2	Dasar-dasar Database Revisi	Budi Santoso	Teknologi	10

Update di Tabel Peminjaman untuk mengupdate status

```

DELIMITER $$
● CREATE PROCEDURE insert_peminjaman(
    pIdSiswa INT,
    pIdBuku INT,
    pTanggalPinjam DATE,
    pTanggalkembali DATE,
    pstatus VARCHAR(20)
)
BEGIN
    INSERT INTO peminjaman (id_siswa, id_buku, tanggal_pinjam, tanggal_kembali, status)
    VALUES (pIdSiswa, pIdBuku, pTanggalPinjam, pTanggalkembali, pstatus);
END $$
DELIMITER ;
● CALL insert_peminjaman(6, 15, '2025-02-01', '2025-02-08', 'Dipinjam');

```

Hasil

	id_peminjaman	id_buku	id_siswa	tanggal_pinjam	tanggal_kembali	status
	11	15	6	2025-02-01	2025-02-08	Dipinjam

Delete di Tabel Siswa


```

153 DELIMITER $$
154 • create procedure delete_Siswa(In new_Id int)
155   Begin
156     DELETE FROM siswa where id_siswa = new_Id;
157   END $$
158 DELIMITER ;
159 • CALL delete_Siswa(15);

```

Hasil

id_siswa	nama	kelas
11	Kartika Sari	XII-RPL
12	Lintang Putri	XII-TKJ
13	Muhammad Rizky	X-RPL
14	Novi Andriana	X-TKJ
NULL	NULL	NULL

Delete di Tabel Buku

```

DELIMITER $$
CREATE PROCEDURE delete_buku(pIdBuku INT)
BEGIN
  DELETE FROM buku WHERE id_buku = pIdBuku;
END $$
DELIMITER ;
CALL delete_buku(15);

```

Hasil

	id_buku	judul_buku	penulis	kategori	stock
	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
	NULL	NULL	NULL	NULL	NULL

Delete di Tabel Peminjaman

```

DELIMITER $$
CREATE PROCEDURE delete_peminjaman(pIdPeminjaman INT)
BEGIN
    DELETE FROM peminjaman WHERE id_peminjaman = pIdPeminjaman;
END $$
DELIMITER ;

select*from peminjaman;
CALL delete_peminjaman(11);

```

Hasil

id_peminjaman	id_buku	id_siswa	tanggal_pinjam	tanggal_kembali	status
7	5	2	2025-01-28	2025-02-04	Dikembalikan
8	8	3	2025-02-02	2025-02-09	Dipinjam
9	10	4	2025-01-30	2025-02-06	Dikembalikan
10	3	5	2025-01-25	2025-02-01	Dikembalikan
NULL	NULL	NULL	NULL	NULL	NULL

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

```

DELIMITER $$
CREATE PROCEDURE tampil_siswa()
BEGIN
    SELECT * FROM siswa;
END $$
DELIMITER ;

CALL tampil_siswa();

```

Tabel Buku

```

DELIMITER $$
CREATE PROCEDURE tampil_buku()
BEGIN
    SELECT * FROM buku;
END $$
DELIMITER ;

CALL tampil_buku();

```

Tabel Peminjaman

```

DELIMITER $$
CREATE PROCEDURE tampil_peminjaman()
BEGIN
    SELECT * FROM peminjaman;
END $$
DELIMITER ;

CALL tampil_peminjaman();

```

7. Stok buku pada saat dipinjam berkurang secara otomatis.

```

CREATE TRIGGER kurangi_stok_buku
AFTER INSERT ON peminjaman
FOR EACH ROW
BEGIN
    UPDATE buku
    SET stokc = stock - 1
    WHERE id_buku = NEW.id_buku;
END //

```

8. Stok buku pada saat dikembalikan bertambah secara otomatis.

```

CREATE TRIGGER tambah_stok_buku
AFTER UPDATE ON peminjaman
FOR EACH ROW
BEGIN
    IF NEW.status = 'Dikembalikan' AND OLD.status != 'Dikembalikan' THEN
        UPDATE buku
        SET stock = stock + 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).

```

CREATE PROCEDURE kembalikan_buku(IN p_id_peminjaman INT)
BEGIN
    UPDATE peminjaman
    SET status = 'Dikembalikan', tanggal_kembali = CURRENT_DATE
    WHERE id_peminjaman = p_id_peminjaman;
    UPDATE buku
    SET stock = stock + 1
    WHERE id_buku = (SELECT id_buku FROM peminjaman WHERE id_peminjaman = p_id_peminjaman);
END $$

```

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

```

CREATE PROCEDURE daftar_siswa_peminjam()
BEGIN
    SELECT DISTINCT s.id_siswa, s.nama_siswa
    FROM siswa s
    JOIN peminjaman p ON s.id_siswa = p.id_siswa;
END $$

```

```

DELIMITER ;
DROP PROCEDURE IF EXISTS daftar_siswa_peminjam;

```

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

```

CREATE PROCEDURE daftar_siswa_peminjam()
> BEGIN
    SELECT DISTINCT s.id_siswa, s.nama
    FROM siswa s
    JOIN peminjaman p ON s.id_siswa = p.id_siswa;
END $$

DELIMITER ;

CALL daftar_siswa_peminjam();

```

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

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

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